**实 验 报 告**

**课程名称** 移动应用系统

**实验项目** 通过网络发送和获取数据

**实验仪器** 台式电脑

**系 别** 计算机科学与技术系

**专 业** 计算机科学与技术

**班级/学号** 计科1905 / 2019010132

**学生姓名** 陈浩

**实验日期** 2021/12/9

**成 绩**

**指导教师** 陈雷

实验三 通过网络发送和获取数据

1. 实验目的：

1、通过本实验，进一步熟悉Android Studio的用法

2、通过本实验，掌握Android系统中网络相关API的使用方法

3、通过本实验，进一步熟悉适配器视图、数据持久化等相关知识的使用。

1. 实验内容及要求：

【注：可以自己设计题目，但需要经过指导教师同意】

【基本要求】设计并实现一个**天气预报程序**，具体要求如下【70分】

1）设计程序界面，允许用户输入城市ID，通过访问在线天气API，获取并显示该城市的天气预报数据（至少要显示省、市、日期、数据更新时间、温度、湿度、PM2.5等数据）。对于非法的城市ID（位数不足9位或不存在对应的城市，可以从返回值中判断），应该给出适当的提示。

|  |
| --- |
| 关于在线天气API的说明  (1)访问https://www.sojson.com/blog/305.html了解关于天气API的详细说明。  (2)天气数据的请求链接为：  <http://t.weather.sojson.com/api/weather/city/101030100>  其中“101030100”为城市ID，对应于天津市。该请求链接的返回结果内容格式说明见<https://www.sojson.com/blog/305.html>。  (3)天气API支持的城市列表包含在city.json文件中，(可以通过如下链接下载：<http://cdn.sojson.com/_city.json?attname=>)，城市列表分为省-市两个级别，其中，省级行政单位的列表记录示例如下:  {  "\_id": 2,  "id": 2,  "pid": 0,  "city\_code": "",  "city\_name": "安徽"  }  市级行政单位的列表记录示例如下：  {  "\_id": 35,  "id": 35,  "pid": 2,  "city\_code": "101220601",  "city\_name": "安庆"  }  可以看出，“city\_code”为空且“pid”为0的列表项表示省级行政单位；“city\_code”不为空的列表项表示市级行政单位，并且市级行政单位的“pid”字段的值是对应的省级行政单位的“id”字段的值。如上面的例子中，安庆市位于安徽省，其pid为2，正好是安徽省对应的数据项的id字段的值。 |

2）使用适当的数据持久化技术（SQLite或SharedPreferences）缓存用户的查询结果（至少缓存最后三次的查询结果），若用户查询的城市在缓存中，则不再访问在线API，直接从缓存中将结果取出显示。

3）在结果展示界面中增加“刷新”按钮，用户点击该按钮后，将重新请求在线数据并更新展示界面中的相关元素。

【扩展要求】在基本要求的基础上，增加如下功能【30分】

1）建立省、市两级行政单位的浏览界面(通过解析城市列表数据city.json实现)，允许用户浏览在线API支持的省、市列表，通过点选的方式选择要查询的城市，并访问在线API获取天气数据加以展示（15分）。

2）允许用户添加关注的城市（可以考虑在天气数据展示页面中添加“关注”按钮，持久化到SharedPreferences或SQLite数据库），应用程序启动时，读取关注的城市列表，在界面中显著位置向用户展示，用户可以点击关注的城市来查看天气数据(15分)。

【评分标准】：

1）实现了基本要求及扩展要求中所要求的程序功能，将得到对应的分数

2）实现功能的同时，应该尽可能保证程序的稳定性、易用性，出现下列情形，将扣去对应的分数

（1）程序崩溃/闪退，扣10分

（2）程序缺少必要的文字提示，扣5分

3）编写程序的过程中，可以参考网上的资料，但必须能够看懂程序代码并回答老师提出的问题，如果无法解释清楚程序代码，扣15分。

4）雷同的实验报告，提交时间较晚的同学成绩为0

1. 程序代码及运行结果：

注：提供完整的程序代码以及运行结果截图

程序代码：

db.City.java

*package* com.example.weather.db;  
  
*import* org.litepal.crud.LitePalSupport;  
  
*public class* City *extends* LitePalSupport {  
  
 *// 存放市级数据  
 private* String cityName;  
 *private* String cityCode;  
 *private* String provinceCode;  
  
 *public* String getCityName() {  
 *return* cityName;  
 }  
  
 *public void* setCityName(String cityName) {  
 *this*.cityName = cityName;  
 }  
  
 *public* String getCityCode() {  
 *return* cityCode;  
 }  
  
 *public void* setCityCode(String cityCode) {  
 *this*.cityCode = cityCode;  
 }  
  
 *public* String getProvinceCode(){  
 *return* provinceCode;  
 }  
  
 *public void* setProvinceCode(String provinceCode) {  
 *this*.provinceCode = provinceCode;  
 }  
}

db.County.java

*package* com.example.weather.db;  
  
*import* org.litepal.crud.LitePalSupport;  
  
*public class* County *extends* LitePalSupport {  
  
 *// 存放县级数据  
 private* String countyName;  
 *private* String countyCode;  
 *private* String cityCode;  
  
  
 *public* String getCountyName() {  
 *return* countyName;  
 }  
  
 *public void* setCountyName(String countyName) {  
 *this*.countyName = countyName;  
 }  
  
 *public* String getCityCode(){  
 *return* cityCode;  
 }  
  
 *public void* setCityCode(String cityCode){  
 *this*.cityCode = cityCode;  
 }  
  
 *public void* setCountyCode(String countyCode){  
 *this*.countyCode = countyCode;  
 }  
  
 *public* String getCountyCode(){  
 *return* countyCode;  
 }  
  
  
}

db.Province.java

*package* com.example.weather.db;  
  
*import* org.litepal.crud.LitePalSupport;  
  
*public class* Province *extends* LitePalSupport {  
  
 *// 存放省级数据  
 private* String provinceName;  
 *private* String provinceCode;  
  
 *public* String getProvinceName() {  
 *return* provinceName;  
 }  
  
 *public void* setProvinceName(String provinceName) {  
 *this*.provinceName = provinceName;  
 }  
  
 *public* String getProvinceCode() { *return* provinceCode; }  
  
 *public void* setProvinceCode(String provinceCode) {  
 *this*.provinceCode = provinceCode;  
 }  
  
}

gson.Weather.java

*package* com.example.weather.gson;  
  
*import* com.google.gson.annotations.*SerializedName*;  
  
*public class* Weather {  
  
 *// 采用注解：  
 @SerializedName*("province")  
 *public* String provinceName;  
  
 *@SerializedName*("city")  
 *public* String cityName;  
  
 *@SerializedName*("adcode")  
 *public* String adcodeName;  
  
 *@SerializedName*("weather")  
 *public* String weatherName;  
  
 *@SerializedName*("temperature")  
 *public* String temperatureName;  
  
 *@SerializedName*("humidity")  
 *public* String humidityName;  
  
 *@SerializedName*("reporttime")  
 *public* String reportTimeName;  
  
}

service.AutoUpdateService.java

*package* com.example.weather.service;  
  
*import* android.app.AlarmManager;  
*import* android.app.PendingIntent;  
*import* android.app.Service;  
*import* android.content.Intent;  
*import* android.content.SharedPreferences;  
*import* android.os.IBinder;  
*import* android.os.SystemClock;  
*import* android.preference.PreferenceManager;  
  
*import* com.example.weather.gson.Weather;  
*import* com.example.weather.util.HttpUtil;  
*import* com.example.weather.util.Utility;  
  
*import* java.io.IOException;  
  
*import* okhttp3.Call;  
*import* okhttp3.Callback;  
*import* okhttp3.Response;  
  
*public class* AutoUpdateService *extends* Service {  
 *// 自动更新服务* @Override  
 *public int* onStartCommand(Intent intent, *int* flags, *int* startId) {  
 *// 自动更新天气信息* updateWeather();  
  
 AlarmManager manager = (AlarmManager) getSystemService(ALARM\_SERVICE);*//书中P469  
 // 8小时的毫秒数  
 int* anHour = 8 \* 60 \* 60 \* 1000;  
 *long* triggerAtTime = SystemClock.elapsedRealtime() + anHour;  
 Intent i = *new* Intent(*this*, AutoUpdateService.*class*);  
 PendingIntent pi = PendingIntent.getService(*this*, 0, i, 0);  
 manager.cancel(pi);  
 manager.set(AlarmManager.ELAPSED\_REALTIME\_WAKEUP, triggerAtTime, pi);  
 *return super*.onStartCommand(intent, flags, startId);  
 }  
  
 @Override  
 *public* IBinder onBind(Intent intent) {  
 *return null*;  
 }  
  
 *// 自动更新天气信息  
 private void* updateWeather(){  
 *// 尝试获取之前保存的缓存* SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences(*this*);  
 String weatherString = prefs.getString("weather", *null*);  
  
 *// 获取当前天气信息 并进行 更新  
 if* (weatherString != *null*) {  
 Weather weather = Utility.handleWeatherResponse(weatherString);  
 *assert* weather != *null*;  
 String weatherId = weather.adcodeName;  
 String weatherUrl = "http://guolin.tech/api/weather?cityid=" + weatherId + "&key=b38826493b8a477eb8c1334f30de6ae2";  
 HttpUtil.sendOkHttpRequest(weatherUrl, *new* Callback() {  
 *// 响应更新* @Override  
 *public void* onResponse(Call call, Response response) *throws* IOException {  
 String responseText = response.body().string();  
 Weather weather = Utility.handleWeatherResponse(responseText);  
 *if* (weather != *null*){  
 SharedPreferences.Editor editor = PreferenceManager.getDefaultSharedPreferences(AutoUpdateService.*this*).edit();  
 editor.putString("weather", responseText);  
 editor.apply();  
 }  
 }  
  
 *// 更新失败报错：* @Override  
 *public void* onFailure(Call call, IOException e) {  
 e.printStackTrace();  
 }  
 });  
 }  
 }  
}

util.HttpUtil.java

*package* com.example.weather.util;  
  
*import* okhttp3.OkHttpClient;  
*import* okhttp3.Request;  
  
*public class* HttpUtil {  
  
 *// 向API发送http请求  
 public static void* sendOkHttpRequest(*final* String address, okhttp3.Callback callback){  
 OkHttpClient client = *new* OkHttpClient();  
 Request request = *new* Request.Builder().url(address).get().build();  
 client.newCall(request).enqueue(callback);  
 }  
}

util.Utility.java

*package* com.example.weather.util;  
  
*import* android.text.TextUtils;  
  
*import* com.google.gson.Gson;  
  
*import* org.json.JSONArray;  
*import* org.json.JSONException;  
*import* org.json.JSONObject;  
  
*import* com.example.weather.db.City;  
*import* com.example.weather.db.County;  
*import* com.example.weather.db.Province;  
*import* com.example.weather.gson.Weather;  
  
*public class* Utility {  
  
 *// 解析和保存服务器所返回的 省级数据  
 public static boolean* handleProvinceResponse(String response){  
 *if* (!TextUtils.isEmpty(response)) {  
 *try* {  
 JSONObject jsonObject = *new* JSONObject(response);  
 JSONArray countryAll = jsonObject.getJSONArray("districts");  
 *for* (*int* i = 0; i < countryAll.length(); i++) {  
 JSONObject countryLeve0 = countryAll.getJSONObject(i);  
 *// 插入省* JSONArray provinceAll = countryLeve0.getJSONArray("districts");  
 *for* (*int* j = 0; j < provinceAll.length(); j++) {  
 JSONObject province1 = provinceAll.getJSONObject(j);  
 String adcode1 = province1.getString("adcode");  
 String name1 = province1.getString("name");  
 Province provinceN = *new* Province();  
 provinceN.setProvinceCode(adcode1);  
 provinceN.setProvinceName(name1);  
 provinceN.save();  
 }  
 *return true*;  
 }  
 }  
 *catch*(JSONException e){  
 e.printStackTrace();  
 }  
 }  
 *return false*;  
 }  
  
 *// 解析和保存服务器所返回的 市级数据  
 public static boolean* handleCityResponse(String response, String provinceCode){  
 *if* (!TextUtils.isEmpty(response)) {  
 *try* {  
 JSONObject jsonObject = *new* JSONObject(response);  
 JSONArray provinceAll = jsonObject.getJSONArray("districts");  
 *for* (*int* i = 0; i < provinceAll.length(); i++) {  
 JSONObject province1 = provinceAll.getJSONObject(i);  
 *// 插入市* JSONArray cityAll = province1.getJSONArray("districts");  
 *for* (*int* j = 0; j < cityAll.length(); j++) {  
 JSONObject city2 = cityAll.getJSONObject(j);  
 String adcode2 = city2.getString("adcode");  
 String name2 = city2.getString("name");  
 City cityN = *new* City();  
 cityN.setCityCode(adcode2);  
 cityN.setCityName(name2);  
 cityN.setProvinceCode(provinceCode);  
 cityN.save();  
 }  
 *return true*;  
 }  
 }  
 *catch*(JSONException e){  
 e.printStackTrace();  
 }  
 }  
 *return false*;  
 }  
  
 *// 解析和保存服务器所返回的 县级数据  
 public static boolean* handleCountyResponse(String response, String cityCode){  
 *if* (!TextUtils.isEmpty(response)) {  
 *try* {  
 JSONObject jsonObject = *new* JSONObject(response);  
 JSONArray cityAll = jsonObject.getJSONArray("districts");  
 *for* (*int* i = 0; i < cityAll.length(); i++) {  
 JSONObject city2 = cityAll.getJSONObject(i);  
 *// 插入县* JSONArray countyAll = city2.getJSONArray("districts");  
 *for* (*int* j = 0; j < countyAll.length(); j++) {  
 JSONObject county3 = countyAll.getJSONObject(j);  
 String adcode3 = county3.getString("adcode");  
 String name3 = county3.getString("name");  
 County countyN = *new* County();  
 countyN.setCountyCode(adcode3);  
 countyN.setCountyName(name3);  
 countyN.setCityCode(cityCode);  
 countyN.save();  
 }  
 *return true*;  
 }  
 }  
 *catch*(JSONException e){  
 e.printStackTrace();  
 }  
 }  
 *return false*;  
 }  
  
 *// 将API返回的 JSON 数据进行解析 并封装为 Weather 类  
 public static* Weather handleWeatherResponse(String response) {  
 *try* {  
 JSONObject jsonObject = *new* JSONObject(response);  
 JSONArray jsonArray = jsonObject.getJSONArray("lives");  
 *for*(*int* i=0; i<jsonArray.length(); i++){  
 JSONObject x = jsonArray.getJSONObject(i);  
 String weatherContent = x.toString();  
 *return new* Gson().fromJson(weatherContent, Weather.*class*);  
 } }*catch* (Exception e) {  
 e.printStackTrace();  
 }  
 *return null*;  
 }  
}

ChooseAreaFragment.java

*package* com.example.weather;  
  
*import* android.content.Intent;  
*import* android.os.Bundle;  
*import* android.view.LayoutInflater;  
*import* android.view.View;  
*import* android.view.ViewGroup;  
*import* android.widget.AdapterView;  
*import* android.widget.ArrayAdapter;  
*import* android.widget.Button;  
*import* android.widget.ListView;  
*import* android.widget.TextView;  
*import* android.widget.Toast;  
  
*import* androidx.annotation.*NonNull*;  
*import* androidx.annotation.*Nullable*;  
*import* androidx.fragment.app.Fragment;  
  
*import* org.jetbrains.annotations.*NotNull*;  
*import* org.litepal.LitePal;  
  
*import* java.io.IOException;  
*import* java.util.ArrayList;  
*import* java.util.List;  
  
*import* com.example.weather.db.City;  
*import* com.example.weather.db.County;  
*import* com.example.weather.db.Province;  
*import* okhttp3.Call;  
*import* okhttp3.Callback;  
*import* okhttp3.Response;  
*import* com.example.weather.util.HttpUtil;  
*import* com.example.weather.util.Utility;  
  
*public class* ChooseAreaFragment *extends* Fragment {  
 *// 设置地区的级别  
 private static final int* LEVEL\_PROVINCE = 0;  
 *private static final int* LEVEL\_CITY = 1;  
 *private static final int* LEVEL\_COUNTY = 2;  
  
 *// 使用datalist列表  
 // 存储保存的地区的 省、市、县和天气 数据  
 private* List<String> dataList = *new* ArrayList<>();  
  
 *private* ArrayAdapter<String> adapter;  
 *private* TextView titleText;  
 *private* Button backButton;  
 *private* ListView listView;  
  
 *// 省级列表  
 private* List<Province> provinceList;  
 *// 市级列表  
 private* List<City> cityList;  
 *// 县级列表  
 private* List<County> countyList;  
 *// 当前等级  
 // 区分 省 市 县  
 private int* currentLevel;  
  
 *// 所选中的省份  
 private* Province selectedProvince;  
 *// 所选中的市区  
 private* City selectedCity;  
  
 *@Nullable  
 @Override  
 public* View onCreateView(*@NonNull* LayoutInflater inflater, *@Nullable* ViewGroup container, *@Nullable* Bundle savedInstanceState) {  
 *// 获得布局填充器* View view = inflater.inflate(R.layout.choose\_area,container,*false*);  
 *// 获得当前布局的各个组件* titleText = view.findViewById(R.id.title\_text);  
 backButton = view.findViewById(R.id.back\_button);  
 listView = view.findViewById(R.id.list\_view);  
  
 adapter = *new* ArrayAdapter<>(getContext(),android.R.layout.simple\_list\_item\_1,dataList);  
 listView.setAdapter(adapter);  
 *return* view;  
 }  
  
 *@Override  
 public void* onActivityCreated(*@Nullable* Bundle savedInstanceState) {  
 *super*.onActivityCreated(savedInstanceState);  
  
 *// 用于侧滑菜单：* listView.setOnItemClickListener(*new* AdapterView.OnItemClickListener() {  
 *@Override  
 public void* onItemClick(AdapterView<?> parent, View view, *int* position, *long* id) {  
 *// 若点击区域的等级为省级  
 if* (currentLevel == LEVEL\_PROVINCE) {  
 *// 获取当前省级地区的位置* selectedProvince = provinceList.get(position);  
 *// 查询数据库或网络资源  
 // 获取当前省份中的市区信息* queryCities();  
 }  
 *// 若点击区域的等级为市级  
 else if* (currentLevel == LEVEL\_CITY) {  
 *// 获取当前省级地区的位置* selectedCity = cityList.get(position);  
 *// 获取当前市区中的县级信息* queryCounties();  
 }  
 *// 若点击区域的等级为县级 (需要最后显示的资源)  
 else if* (currentLevel == LEVEL\_COUNTY) {  
 *// 获得当前县级的名称和代码* String countyCode = countyList.get(position).getCountyCode();  
 String countyName = countyList.get(position).getCountyName();  
  
 *// 判断当前执行的活动是否为 主活动类型  
 // 即是否正在选取 市、区、县  
 if* (getActivity() *instanceof* MainActivity) {  
 *// 准备启动该活动* Intent intent = *new* Intent(getActivity(), WeatherActivity.*class*);  
 *// 将当前最后一级的 县级名称和代码 进行保存 并传入活动* intent.putExtra( "adcode",countyCode);  
 intent.putExtra("city",countyName);  
 startActivity(intent);  
 getActivity().finish();  
 }  
 *// 判断当前执行的活动是否为 查看天气的活动类型  
 else if* (getActivity() *instanceof* WeatherActivity) {  
 WeatherActivity activity = (WeatherActivity) getActivity();  
 *// 关闭左侧滑动菜单* activity.drawerLayout.closeDrawers();  
 *// 启用下拉刷新进度条* activity.swipeRefresh.setRefreshing(*true*);  
 *// 显示当前 县级 的天气情况* activity.requestWeather(countyCode);  
 }  
 }  
 }  
 });  
  
 *// 返回主页按钮：* backButton.setOnClickListener(*new* View.OnClickListener() {  
 *@Override  
 public void* onClick(View v) {  
 *// 判断当前页面的等级  
 if* (currentLevel == LEVEL\_COUNTY) {  
 *// 若当前页面停留在 县级  
 // 则返回至上一级的 市级 页面* queryCities();  
 }  
 *else if* (currentLevel == LEVEL\_CITY) {  
 *// 若当前页面停留在 市级  
 // 则返回至上一级的 省级 页面* queryProvinces();  
 }  
 }  
 });  
  
 *// 默认停留在 省级 页面* queryProvinces();  
 }  
  
 *// 查询所有省份  
 // 优先从当前数据库查询  
 // 若不存在则再去服务器上查询  
 private void* queryProvinces(){  
 titleText.setText("中 国 地 区 省 份");  
 *// 隐藏返回按钮* backButton.setVisibility(View.GONE);  
  
 *// 当前数据库(LitePal)中查询：* provinceList = LitePal.findAll(Province.*class*);  
 *if* (provinceList.size() > 0) {  
 dataList.clear();  
 *// 将datalist更新为中国的所有省份  
 for* (Province province : provinceList) {  
 dataList.add(province.getProvinceName());  
 }  
 *// 刷新ListView (显示更新后的内容)* adapter.notifyDataSetChanged();  
 listView.setSelection(0);  
 *// 将页面等级调整为上一级* currentLevel = LEVEL\_PROVINCE;  
 }  
 *// 若当前数据库没有数据  
 // 则跳调用API进行访问  
 else*{  
 String address = "https://restapi.amap.com/v3/config/district?keywords=中国&subdistrict=1&key=562a75a2243ea6a24389af6f5f954388";  
 *// 查询 省份 信息* queryFromServer(address,"province");  
 }  
 }  
  
 *// 查询所有市区  
 // 优先从当前数据库查询  
 // 若不存在则再去服务器上查询  
 private void* queryCities() {  
 titleText.setText(selectedProvince.getProvinceName());  
 *// 显示返回按钮* backButton.setVisibility(View.VISIBLE);  
  
 *// 需要根据特定 省份 来查询 市区* cityList = LitePal.where("provinceCode = ?",  
 String.valueOf(selectedProvince.getProvinceCode())).find(City.*class*);  
 *// 当前数据库(LitePal)中查询： (以下操作同 queryProvinces())  
 if* (cityList.size() > 0) {  
 dataList.clear();  
 *for* (City city : cityList) {  
 dataList.add(city.getCityName());  
 }  
 adapter.notifyDataSetChanged();  
 listView.setSelection(0);  
 currentLevel = LEVEL\_CITY;  
 }  
 *else* {  
 *// 需要获取先前保存的省份信息* String provinceName = selectedProvince.getProvinceName();  
 String address = "https://restapi.amap.com/v3/config/district?keywords="+provinceName+"&subdistrict=1&key=c1894e9fcaf35e9fceabe9afaf40d45f";  
 *// 查询 市区 信息* queryFromServer(address,"city");  
 }  
 }  
  
 *// (具体方法同上)  
 // 查询所有乡县  
 // 优先从当前数据库查询  
 // 若不存在则再去服务器上查询  
 private void* queryCounties() {  
 titleText.setText(selectedCity.getCityName());  
 backButton.setVisibility(View.VISIBLE);  
  
 countyList = LitePal.where("cityCode=?",  
 String.valueOf(selectedCity.getCityCode())).find(County.*class*);  
 *if* (countyList.size() > 0) {  
 dataList.clear();  
 *for* (County county:countyList){  
 dataList.add(county.getCountyName());  
 }  
 adapter.notifyDataSetChanged();  
 listView.setSelection(0);  
 currentLevel = LEVEL\_COUNTY;  
 }  
 *else* {  
 String cityName = selectedCity.getCityName();  
 String address = "https://restapi.amap.com/v3/config/district?keywords="+cityName+"&subdistrict=1&key=c1894e9fcaf35e9fceabe9afaf40d45f";  
 queryFromServer(address,"county");  
 }  
 }  
  
  
 *// 从传入的API服务器中获取 省 市 县 的数据  
 private void* queryFromServer(String address, *final* String type){  
 HttpUtil.sendOkHttpRequest(address, *new* Callback() {  
 *@Override  
 public void* onFailure(*@NotNull* Call call, *@NotNull* IOException e) {  
 getActivity().runOnUiThread(*new* Runnable() {  
 *@Override  
 public void* run() {  
 Toast.makeText(getContext(), "加载API失败", Toast.LENGTH\_SHORT).show();  
 }  
 });  
 }  
 *@Override  
 public void* onResponse(*@NotNull* Call call, *@NotNull* Response response) *throws* IOException {  
 String responseText = response.body().string();  
 *boolean* result = *false*;  
 *// 根据对应的 Key 值进行查询和传送不同的数据  
 if*("province".equals(type)) {  
 result = Utility.handleProvinceResponse(responseText);  
 }  
 *else if*("city".equals(type)) {  
 result = Utility.handleCityResponse(responseText, selectedProvince.getProvinceCode());  
 }  
 *else if*("county".equals(type)) {  
 result = Utility.handleCountyResponse(responseText, selectedCity.getCityCode());  
 }  
 *// 若查询成功：  
 // 则调用活动中的方法 将其保存到数据库中 方便下次使用  
 if*(result){  
 getActivity().runOnUiThread(*new* Runnable() {  
 *@Override  
 public void* run() {  
 *if*("province".equals(type)) {  
 queryProvinces();  
 }  
 *else if*("county".equals(type)) {  
 queryCities();  
 }  
 *else if*("county".equals(type)) {  
 queryCounties();  
 }  
 }  
 });  
 }  
 }  
 });  
 }  
}

MainActivity.java

*package* com.example.weather;  
  
*import* android.content.Context;  
*import* android.content.Intent;  
*import* android.content.SharedPreferences;  
*import* android.os.Bundle;  
*import* android.view.View;  
*import* android.widget.Button;  
*import* android.widget.EditText;  
*import* android.widget.Toast;  
  
*import* androidx.appcompat.app.AppCompatActivity;  
  
*public class* MainActivity *extends* AppCompatActivity {  
 *// 查找按钮  
 private* Button searchButton;  
 *// 通过县级代码 查询天气  
 private* EditText chengShi;  
 *// 我的关注按钮  
 private* Button myConcern;  
  
 *@Override  
 protected void* onCreate(*final* Bundle savedInstanceState) {  
 *super*.onCreate(savedInstanceState);  
 *// 添加组件* setContentView(R.layout.activity\_main);  
 chengShi = findViewById(R.id.chengshi\_text);  
 searchButton = findViewById(R.id.search\_button);  
 myConcern = findViewById(R.id.concern\_text);  
  
 *// 搜索按钮监听* searchButton.setOnClickListener(*new* View.OnClickListener() {  
 *@Override  
 public void* onClick(View v) {  
 String searchCountyCode = String.valueOf(chengShi.getText());  
 *// 高德天气中 城市代码为 6位  
 if*(searchCountyCode.length() != 6) {  
 Toast.makeText(MainActivity.*this*,"城市ID长度为6位!",Toast.LENGTH\_LONG).show();  
 }  
 *else* {  
 Intent intent = *new* Intent(MainActivity.*this*, WeatherActivity.*class*);  
 intent.putExtra("adcode",searchCountyCode);  
 startActivity(intent);  
 }  
 }  
 });  
  
 *// 我的关注按钮监听* myConcern.setOnClickListener(*new* View.OnClickListener() {  
 *@Override  
 public void* onClick(View v) {  
 Intent intent = *new* Intent(MainActivity.*this*,MyConcernList.*class*);  
 startActivity(intent);  
 }  
 });  
  
 *// SharedPreferences 存储数据* SharedPreferences pres = getSharedPreferences(String.valueOf(*this*), Context.MODE\_PRIVATE);  
 *if* (pres.getString("weather", *null*) != *null*){  
 Intent intent = *new* Intent(*this*,WeatherActivity.*class*);  
 startActivity(intent);  
 finish();  
 }  
 }  
}

MyConcernList.java

*package* com.example.weather;  
  
*import* android.content.Intent;  
*import* android.database.Cursor;  
*import* android.database.sqlite.SQLiteDatabase;  
*import* android.os.Bundle;  
*import* android.view.View;  
*import* android.widget.AdapterView;  
*import* android.widget.ArrayAdapter;  
*import* android.widget.Button;  
*import* android.widget.ListView;  
  
*import* androidx.appcompat.app.AppCompatActivity;  
  
*import* java.util.ArrayList;  
*import* java.util.List;  
  
*import static* com.example.weather.MyDBhelper.DB\_NAME;  
  
*public class* MyConcernList *extends* AppCompatActivity {  
 ArrayAdapter simpleAdapter;  
 ListView MyConcernList;  
 *private* Button goBack;  
 *private* List<String> city\_nameList = *new* ArrayList<>();  
 *private* List<String> city\_codeList = *new* ArrayList<>();  
  
 *// 将获取的数据填充到数据库中  
 // 初始化列表  
 private void* InitConcern() {  
 MyDBhelper dbHelper = *new* MyDBhelper(*this*, DB\_NAME, *null*, 1);  
 SQLiteDatabase db = dbHelper.getWritableDatabase();  
 Cursor cursor = db.rawQuery("select \* from Concern",*null*);  
 *while*(cursor.moveToNext()){  
 String city\_code = cursor.getString(cursor.getColumnIndex("city\_code"));  
 String city\_name = cursor.getString(cursor.getColumnIndex("city\_name"));  
 city\_codeList.add(city\_code);  
 city\_nameList.add(city\_name);  
 }  
 }  
  
 *// 刷新列表 (先清除 再添加)  
 public void* RefreshList(){  
 city\_nameList.removeAll(city\_nameList);  
 city\_codeList.removeAll(city\_codeList);  
 simpleAdapter.notifyDataSetChanged();  
 MyDBhelper dbHelper = *new* MyDBhelper(*this*,DB\_NAME,*null*,1);  
 SQLiteDatabase db = dbHelper.getWritableDatabase();  
 Cursor cursor = db.rawQuery("select \* from Concern",*null*);  
 *while*(cursor.moveToNext()){  
 String city\_code = cursor.getString(cursor.getColumnIndex("city\_code"));  
 String city\_name = cursor.getString(cursor.getColumnIndex("city\_name"));  
 city\_codeList.add(city\_code);  
 city\_nameList.add(city\_name);  
 }  
 }  
  
 @Override  
 *protected void* onStart(){  
 *super*.onStart();  
 RefreshList();  
 }  
 @Override  
 *protected void* onCreate(Bundle savedInstanceState) {  
 *super*.onCreate(savedInstanceState);  
 setContentView(R.layout.myconcern\_list);  
 MyConcernList = findViewById(R.id.MyConcernList);  
 goBack = findViewById(R.id.goback\_btn);  
  
 InitConcern();  
  
 *// 设置适配器* simpleAdapter = *new* ArrayAdapter(MyConcernList.*this*,android.R.layout.simple\_list\_item\_1,city\_nameList);  
 MyConcernList.setAdapter(simpleAdapter);  
  
 *// ArrayList点击事件  
 // 即点击县级能够跳转至天气界面* MyConcernList.setOnItemClickListener(*new* AdapterView.OnItemClickListener(){  
 @Override  
 *public void* onItemClick(AdapterView<?> parent, View view , *int* position , *long* id){  
 String tran = city\_codeList.get(position);  
 Intent intent = *new* Intent(MyConcernList.*this*, WeatherActivity.*class*);  
 intent.putExtra("adcode", tran);  
 startActivity(intent);  
 }  
 });  
  
 *// 返回按钮* goBack.setOnClickListener(*new* View.OnClickListener(){  
 @Override  
 *public void* onClick(View v) {  
 *// 启动主活动 (即返回主页面)* Intent intent = *new* Intent(MyConcernList.*this*, MainActivity.*class*);  
 startActivity(intent);  
 finish();  
 }  
 });  
 }  
}

MyDBhelper.java

*package* com.example.weather;  
  
*import* android.content.Context;  
*import* android.database.sqlite.SQLiteDatabase;  
*import* android.database.sqlite.SQLiteOpenHelper;  
  
*public class* MyDBhelper *extends* SQLiteOpenHelper {  
 *public static final* String DB\_NAME="Minakami.db";  
 *public static final int* VERSION=1;  
 *public static final* String TABLE\_NAME="Concern";  
 *//对数据库进行创建  
 public static final* String CREATE\_CONCERN = "create table Concern("  
 + "city\_code String primary key not null,"  
 + "city\_name String not null)";  
  
 *public* MyDBhelper(Context context, String name, SQLiteDatabase.CursorFactory factory, *int* version){  
 *super*(context,name,factory,version);  
 }  
  
 *@Override  
 public void* onCreate(SQLiteDatabase db){  
 db.execSQL(CREATE\_CONCERN);  
 }  
 *@Override  
 public void* onUpgrade(SQLiteDatabase db, *int* oldVersion, *int* newVersion){  
  
 }  
}

WeatherActivity.java

*package* com.example.weather;  
  
*import* androidx.appcompat.app.AppCompatActivity;  
*import* androidx.core.view.GravityCompat;  
*import* androidx.drawerlayout.widget.DrawerLayout;  
*import* androidx.swiperefreshlayout.widget.SwipeRefreshLayout;  
  
*import* android.annotation.*SuppressLint*;  
*import* android.content.ContentValues;  
*import* android.content.Context;  
*import* android.content.Intent;  
*import* android.content.SharedPreferences;  
*import* android.database.sqlite.SQLiteDatabase;  
*import* android.graphics.Color;  
*import* android.os.Build;  
*import* android.os.Bundle;  
*import* android.preference.PreferenceManager;  
*import* android.view.View;  
*import* android.widget.Button;  
*import* android.widget.ImageView;  
*import* android.widget.ScrollView;  
*import* android.widget.TextView;  
*import* android.widget.Toast;  
  
*import* java.io.IOException;  
*import* java.util.Objects;  
  
*import* com.bumptech.glide.Glide;  
*import* com.example.weather.gson.Weather;  
*import* okhttp3.Call;  
*import* okhttp3.Callback;  
*import* okhttp3.Response;  
  
*import* com.example.weather.util.HttpUtil;  
*import* com.example.weather.util.Utility;  
  
*import static* com.example.weather.MyDBhelper.DB\_NAME;  
*import static* com.example.weather.MyDBhelper.TABLE\_NAME;  
  
*public class* WeatherActivity *extends* AppCompatActivity {  
  
 *public* DrawerLayout drawerLayout;  
 *// title 标签  
 private* Button navButton;  
 *// 关注按钮  
 private* Button concern;  
 *// 取消关注按钮  
 private* Button concealConcern;  
 *// 返回按钮  
 private* Button goBack;  
 *// 刷新按钮  
 private* Button refresh;  
 *// 下拉刷新  
 public* SwipeRefreshLayout swipeRefresh;  
 *// ListView滚动  
 private* ScrollView weatherLayout;  
 *// 省级  
 private* TextView provinceText;  
 *// 市级  
 private* TextView cityText;  
 *// 天气情况  
 private* TextView weatherText;  
 *// 温度  
 private* TextView temperatureText;  
 *// 湿度  
 private* TextView humidityText;  
 *// 实时时间  
 private* TextView reportTimeText;  
  
 @SuppressLint("ResourceAsColor")  
 @Override  
 *protected void* onCreate(Bundle savedInstanceState) {  
 *super*.onCreate(savedInstanceState);  
  
 *if*(Build.VERSION.SDK\_INT >= 21){  
 View decorView = getWindow().getDecorView();  
 *// 当前布局将显示在Title之上* decorView.setSystemUiVisibility(View.SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN | View.SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE);  
 *// 将状态栏设置成透明色* getWindow().setStatusBarColor(Color.TRANSPARENT);  
 }  
  
 *// 获得活动组件* setContentView(R.layout.activity\_weather);  
 weatherLayout = findViewById(R.id.weather\_layout);  
 swipeRefresh = findViewById(R.id.swipe\_refresh);  
 provinceText = findViewById(R.id.province\_text);  
 cityText = findViewById(R.id.city\_text);  
 weatherText = findViewById(R.id.weather\_text);  
 temperatureText = findViewById(R.id.temperature\_text);  
 humidityText = findViewById(R.id.humidity\_text);  
 reportTimeText = findViewById(R.id.reporttime\_text);  
 drawerLayout = findViewById(R.id.drawer\_layout);  
 navButton = findViewById(R.id.nav\_button);  
 concern = findViewById(R.id.concern);  
 concealConcern = findViewById(R.id.concealConcern);  
 goBack = findViewById(R.id.goBack);  
 refresh = findViewById(R.id.refresh);  
  
 *// 获取数据库中的缓存* SharedPreferences prefs = getSharedPreferences(String.valueOf(*this*), Context.MODE\_PRIVATE);  
 String adcodeString = prefs.getString("weather",*null*);  
 *final* String countyCode;  
 *final* String countyName;  
  
 *// 若存在缓存 (即数据库中有数据)  
 // 则可以直接进行天气解析  
 if* (adcodeString != *null*) {  
 Weather weather = Utility.handleWeatherResponse(adcodeString);  
 *assert* weather != *null*;  
 countyCode = weather.adcodeName;  
 countyName = weather.cityName;  
 *// 最后显示天气的具体信息* showWeatherInfo(weather);  
 }  
 *// 若不存在缓存 (即数据库中没有数据)  
 else* {  
 *// 则将需要保存的值设置 Key 值  
 // 并调用API进行查找和保存* countyCode = getIntent().getStringExtra("adcode");  
 countyName = getIntent().getStringExtra("city");  
 weatherLayout.setVisibility(View.INVISIBLE);  
 requestWeather(countyCode);  
 }  
  
 *// 下拉刷新* swipeRefresh.setOnRefreshListener(*new* SwipeRefreshLayout.OnRefreshListener(){*//下拉进度条监听器* @Override  
 *public void* onRefresh() {  
 *// 每次刷新时都进行天气的实时更新* requestWeather(countyCode);  
 }  
 });  
  
 *// title 标签按钮* navButton.setOnClickListener(*new* View.OnClickListener(){  
 @Override  
 *public void* onClick(View v){  
 *// 开启侧滑菜单 (显示内容为最开始的主页内容)* drawerLayout.openDrawer(GravityCompat.START);  
 }  
 });  
  
 *// 关注按钮* concern.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View v) {  
 *// 将当前 县级名称 和 代码 加入数据库中* MyDBhelper dbHelper = *new* MyDBhelper(WeatherActivity.*this*, DB\_NAME, *null*, 1);  
 SQLiteDatabase db = dbHelper.getWritableDatabase();  
 ContentValues values = *new* ContentValues();  
 values.put("city\_code", countyCode);  
 values.put("city\_name", countyName);  
 db.insert(TABLE\_NAME, *null*, values);  
 Toast.makeText(WeatherActivity.*this*, "关注成功！", Toast.LENGTH\_LONG).show();  
 }  
 });  
  
 *// 取消关注按钮* concealConcern.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View v) {  
 *// 将当前 县及名称 和 代码 从数据库中删除* MyDBhelper dbHelper = *new* MyDBhelper(WeatherActivity.*this*, DB\_NAME, *null*, 1);  
 SQLiteDatabase db = dbHelper.getWritableDatabase();  
 db.delete(TABLE\_NAME,"city\_code=?",*new* String[]{String.valueOf(countyCode)});  
 Toast.makeText(WeatherActivity.*this*, "取消关注成功！", Toast.LENGTH\_LONG).show();  
 }  
 });  
  
 *// 返回按钮* goBack.setOnClickListener(*new* View.OnClickListener(){  
 @Override  
 *public void* onClick(View v) {  
 *// 启动主活动 (即返回主页面)* Intent intent = *new* Intent(WeatherActivity.*this*, MainActivity.*class*);  
 startActivity(intent);  
 finish();  
 }  
 });  
  
 *// 刷新按钮 (重新获取天气信息)* refresh.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View v) {  
 requestWeather(countyCode);  
 }  
 });  
 }  
  
 *// 获取天气主要信息  
 public void* requestWeather(*final* String adCode) {  
 *// 调用 天气API 获取 Http请求* String weatherUrl = "https://restapi.amap.com/v3/weather/weatherInfo?city=" + adCode + "&key=562a75a2243ea6a24389af6f5f954388";  
 HttpUtil.sendOkHttpRequest(weatherUrl, *new* Callback() {  
 @Override  
 *public void* onResponse(Call call, Response response) *throws* IOException {  
 *final* String responseText = Objects.requireNonNull(response.body()).string();  
 *final* Weather weather = Utility.handleWeatherResponse(responseText);  
  
 runOnUiThread(*new* Runnable() {  
 @Override  
 *public void* run() {  
 *// 获取API解析成果  
 if* (weather != *null*) {  
 *// 使用 sharePreferences 将获取的天气信息存储* SharedPreferences.Editor editor = getSharedPreferences(String.valueOf(*this*),MODE\_PRIVATE).edit();  
 editor.putString("weather", responseText);  
 editor.apply();  
 showWeatherInfo(weather);  
 }  
 *else* {  
 Toast.makeText(WeatherActivity.*this*, "获取天气信息失败,城市ID不存在，请重新输入！", Toast.LENGTH\_SHORT).show();  
 }  
 *// 停止下拉刷新动画* swipeRefresh.setRefreshing(*false*);  
 }  
 });  
 }  
  
 *// 响应失败：* @Override  
 *public void* onFailure(Call call, IOException e) {  
 e.printStackTrace();  
 runOnUiThread(*new* Runnable() {  
 @Override  
 *public void* run() {  
 Toast.makeText(WeatherActivity.*this*, "获取天气信息失败", Toast.LENGTH\_SHORT).show();  
 *// 停止下拉刷新动画* swipeRefresh.setRefreshing(*false*);  
 }  
 });  
 }  
 });  
 }  
  
 *// 具体天气显示  
 private void* showWeatherInfo(Weather weather) {  
 provinceText.setText(weather.provinceName);  
 cityText.setText(weather.cityName);  
 weatherText.setText("天气: " + weather.weatherName);  
 temperatureText.setText("温度: " + weather.temperatureName + "℃");  
 humidityText.setText("湿度: " + weather.humidityName + "%");  
 reportTimeText.setText(weather.reportTimeName);  
 weatherLayout.setVisibility(View.VISIBLE);  
 }  
}

布局代码略

运行截图：

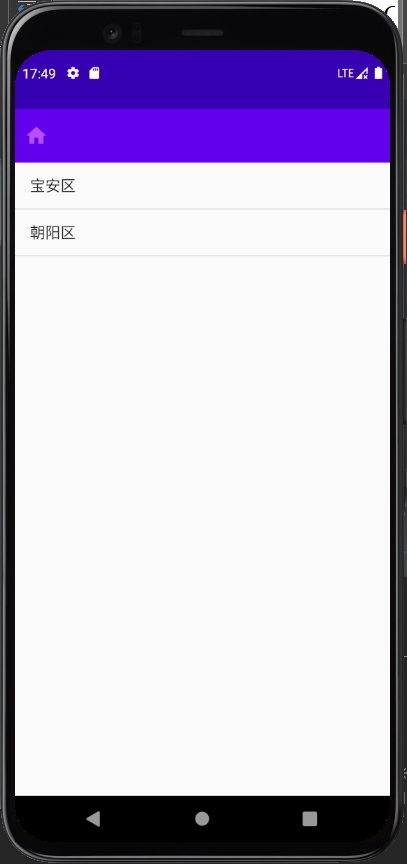




















仓库：

<https://github.com/Minakami-Yuuki/Assignments>

1. 实验总结：

本次实验要求完成天气预报系统，其主要功能需要包含：对省级、市级、区级的选择，并在选择区级后进行天气的各项指标显示，如天气情况、温度、湿度等，在天气界面可以选择关注城市，进行下一次的直接访问。

由于原API已失效，故本次实验采用了高德天气API。

本次实验难点：如何获取虚拟机的网络连接、如何连接到API并调用API对天气情况进行获取、如何设置省级、市级、区级的各个层次跳转、如何进行对城市的关注等。

本次实验，我采用了单帧布局、相对布局和线性布局，在主页面上采用碎片化，引入各个页面的点击跳转。在其余页面上分别设置ListView进行对省级、市级、区级的滑动显示。在实现代码中，首先在MyDBhelper.java内连接数据库，创建表，进行对城市名称和城市代码的存储；在ChooseAreaFragment.java中进行对省级、市级、区级的页面判断和跳转判断，并在最后添加了对天气的获取；在WeatherActivity.java中，对主要活动进行设置，设置4个按钮分别对应至关注页面、刷新当前页面和返回主页面。同时设置了下拉刷新和左侧滚动菜单进行对数据的实时更新。在最后采用了SharePreferences进行对当前城市名称和代码的保存，方面下一次直接可以访问而不用在访问网络API。

经过本次实验，我了解了更多的关于Android开发的知识，并知道了在Android中是如何进行网络配置和连接，希望接下来能够了解更多的Android开发知识。