中山大學

SUN YAT-SEN UNIVERSITY

《手机平台应用开发》

数据存储（二）

学院名称：数据科学与计算机学院

成　　员：陈伟宸

时　　间：2016 年 11 月 25 日

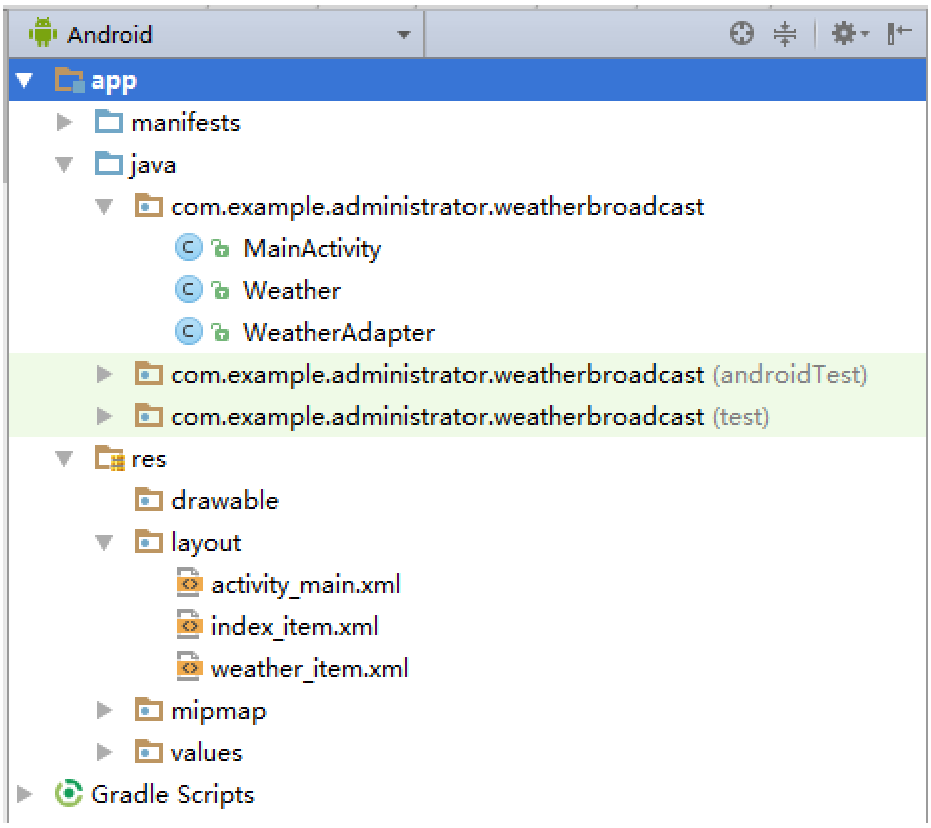
一、实验环境

操作系统：Windows 10

IDE：Android Studio 2.2.2

二、实验过程

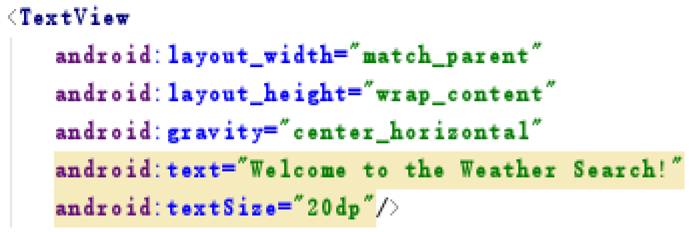
1. 新建一个项目，新建 java 文件和 xml 文件如下：



```
Android
▼ app
    ▶ manifests
    ▼ java
        ▼ com.example.administrator.weatherbroadcast
            MainActivity
            Weather
            WeatherAdapter
        ▶ com.example.administrator.weatherbroadcast (androidTest)
        ▶ com.example.administrator.weatherbroadcast (test)
    ▼ res
        drawable
        ▼ layout
            activity_main.xml
            index_item.xml
            weather_item.xml
        ▶ mipmap
        ▶ values
▶ Gradle Scripts
```

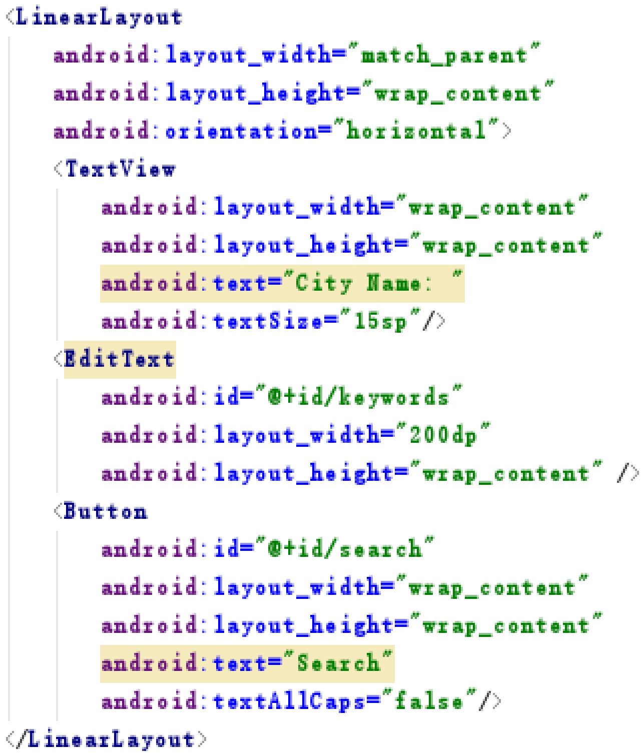
2. 设计 MainActivity 的布局

（1）标题



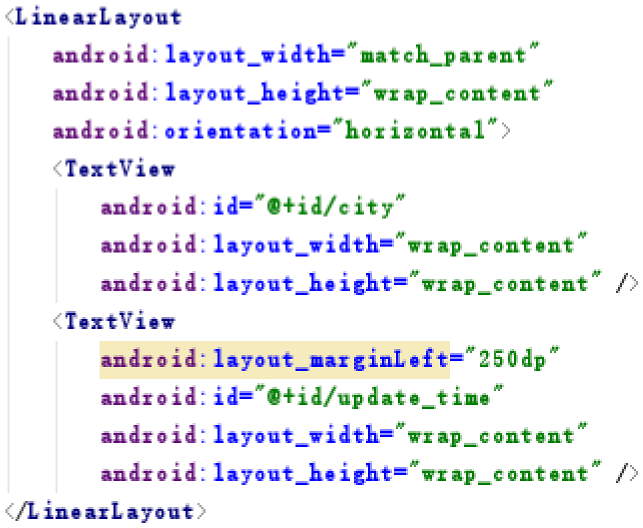
```
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center_horizontal"
    android:text="Welcome to the Weather Search!"
    android:textSize="20dp"/>
```

（2）搜索栏



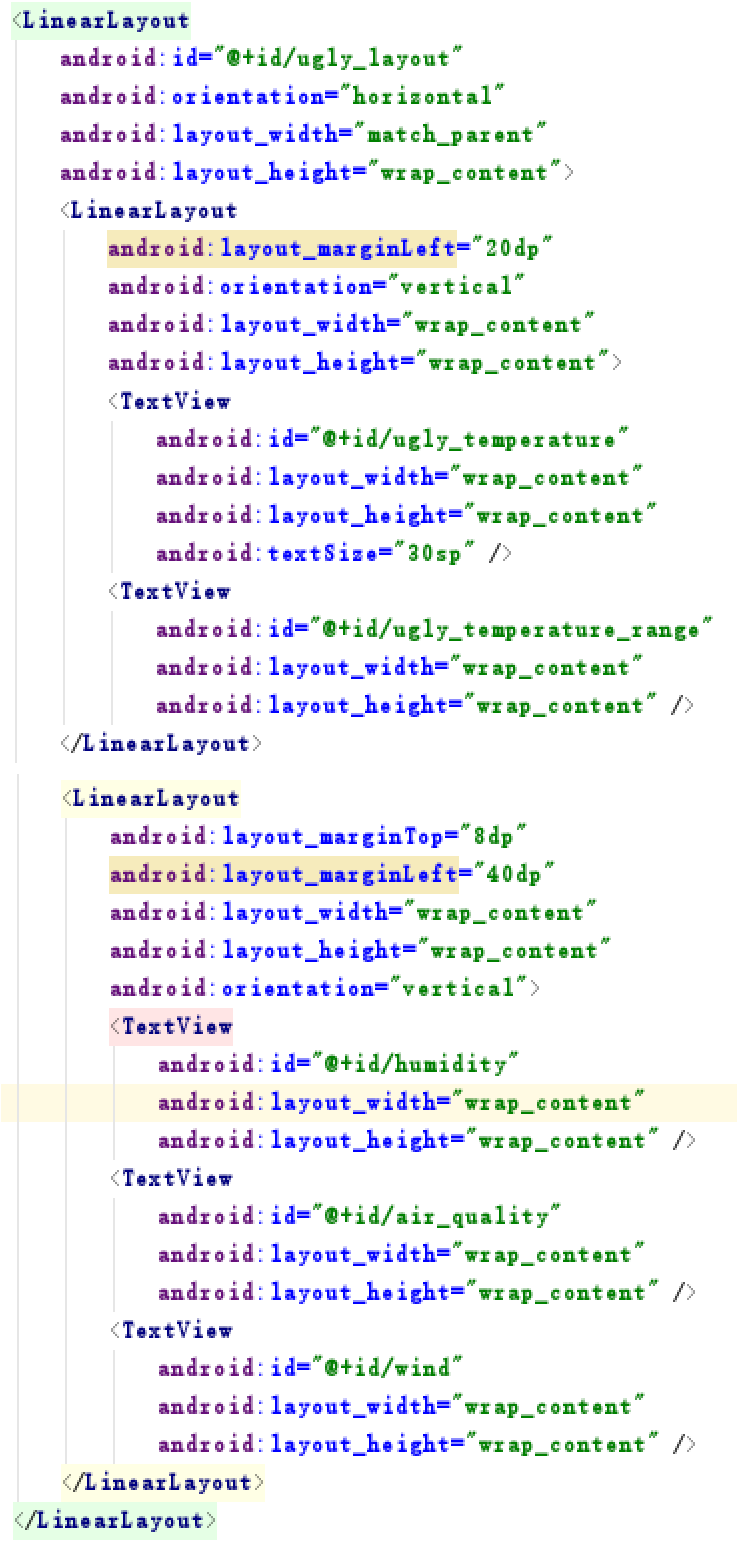
```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="City Name: "
        android:textSize="15sp"/>
    <EditText
        android:id="@+id/keywords"
        android:layout_width="200dp"
        android:layout_height="wrap_content" />
    <Button
        android:id="@+id/search"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Search"
        android:textAllCaps="false"/>
</LinearLayout>
```

（3）城市信息和最后更新时间



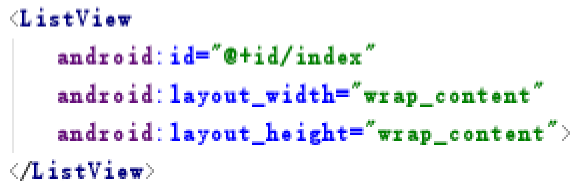
```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">
    <TextView
        android:id="@+id/city"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <TextView
        android:layout_marginLeft="250dp"
        android:id="@+id/update_time"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```

（4）温度、风力、湿度等信息



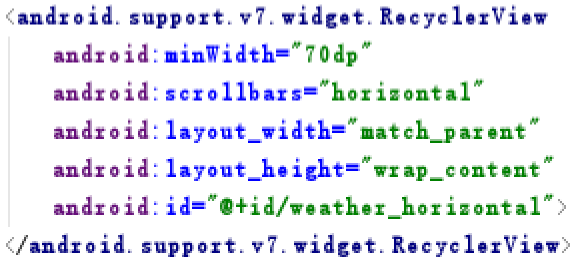
```
<LinearLayout
    android:id="@+id/ugly_layout"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <LinearLayout
        android:layout_marginLeft="20dp"
        android:orientation="vertical"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
        <TextView
            android:id="@+id/ugly_temperature"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="30sp" />
        <TextView
            android:id="@+id/ugly_temperature_range"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </LinearLayout>

    <LinearLayout
        android:layout_marginTop="8dp"
        android:layout_marginLeft="40dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <TextView
            android:id="@+id/humidity"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
        <TextView
            android:id="@+id/air_quality"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
        <TextView
            android:id="@+id/wind"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
    </LinearLayout>
</LinearLayout>
```

（5）紫外线指数、穿衣指数等



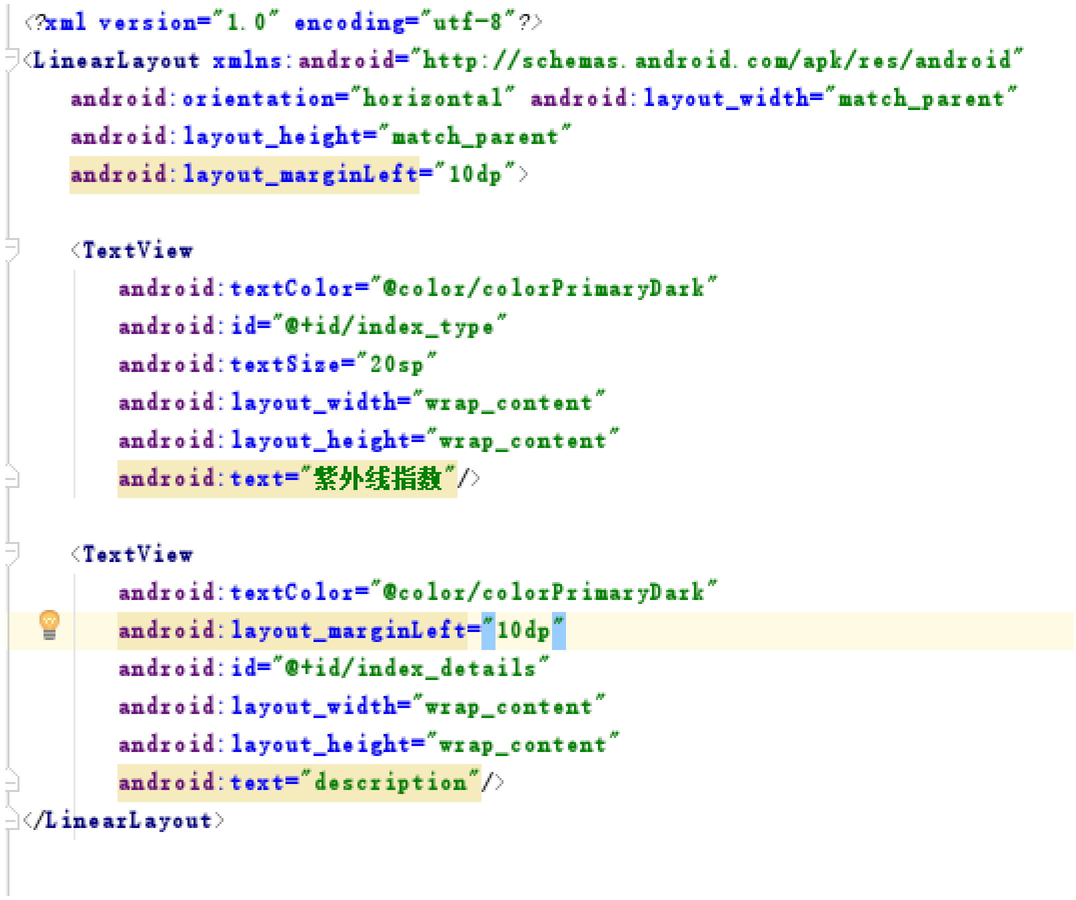
```xml
<ListView
    android:id="@+id/index"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
</ListView>
```

（6）未来几天的天气简要信息



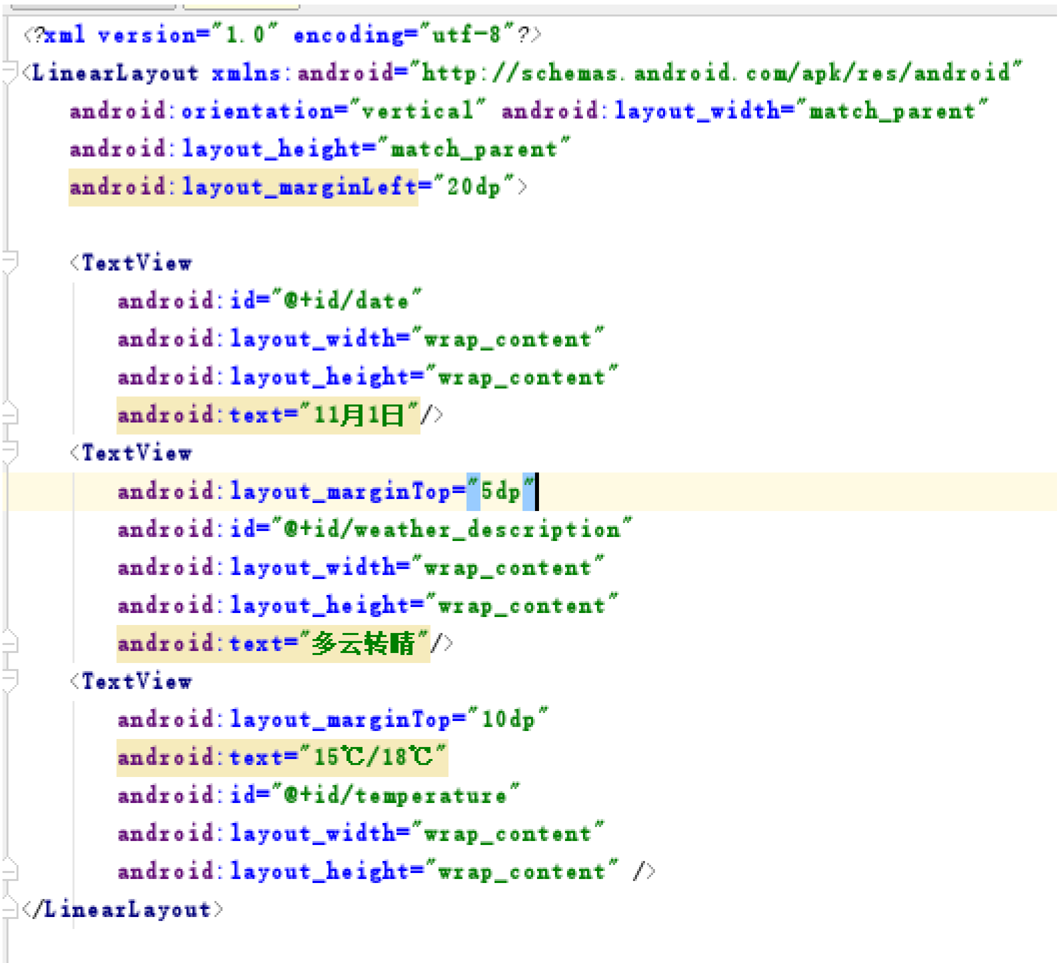
```xml
<android.support.v7.widget.RecyclerView
    android:minWidth="70dp"
    android:scrollbars="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/weather_horizontal">
</android.support.v7.widget.RecyclerView>
```

3. 设计紫外线指数等信息的样式



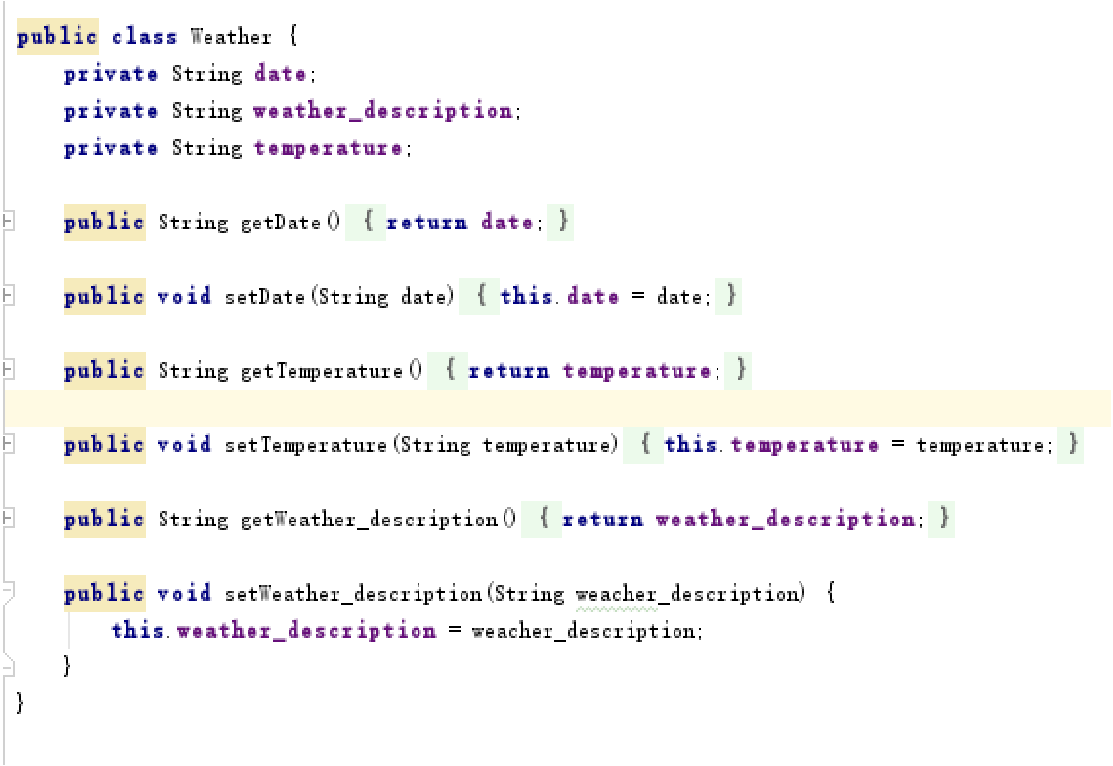
```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="10dp">

    <TextView
        android:textColor="@color/colorPrimaryDark"
        android:id="@+id/index_type"
        android:textSize="20sp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="紫外线指数"/>

    <TextView
        android:textColor="@color/colorPrimaryDark"
        android:layout_marginLeft="10dp"
        android:id="@+id/index_details"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="description"/>
</LinearLayout>
```

4. 设计天气简要信息的样式



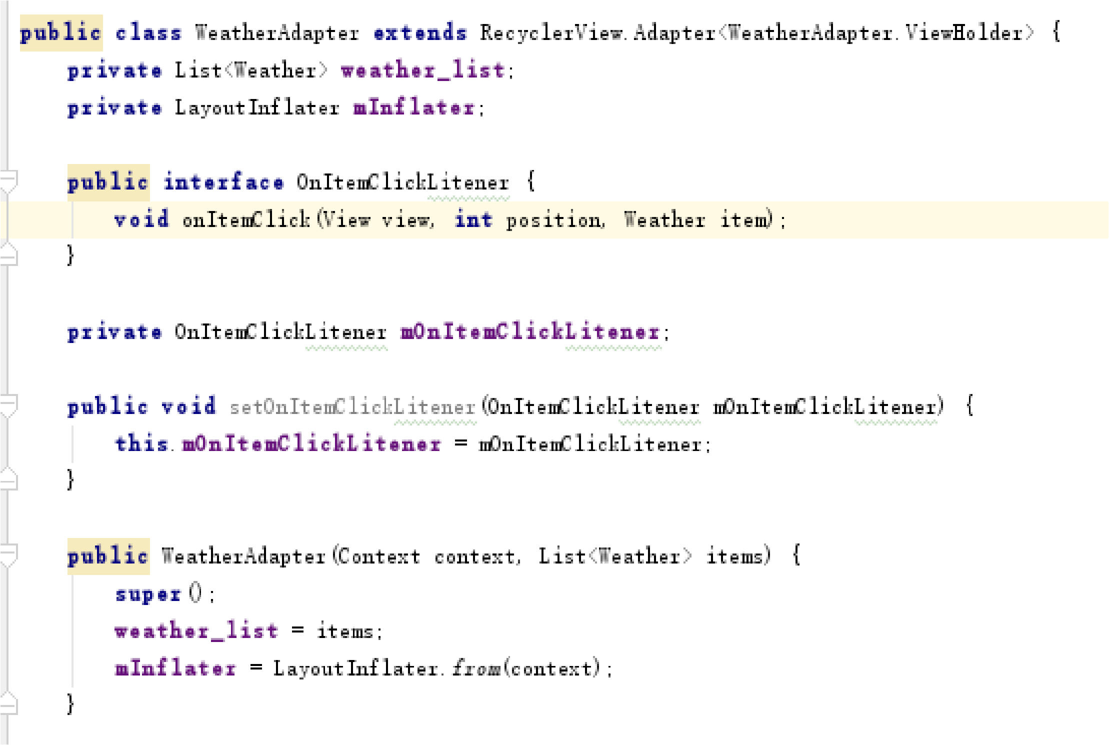
```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginLeft="20dp">

    <TextView
        android:id="@+id/date"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="11月1日"/>
    <TextView
        android:layout_marginTop="5dp"
        android:id="@+id/weather_description"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="多云转晴"/>
    <TextView
        android:layout_marginTop="10dp"
        android:text="15℃/18℃"
        android:id="@+id/temperature"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```

5. 构造 Weather 类

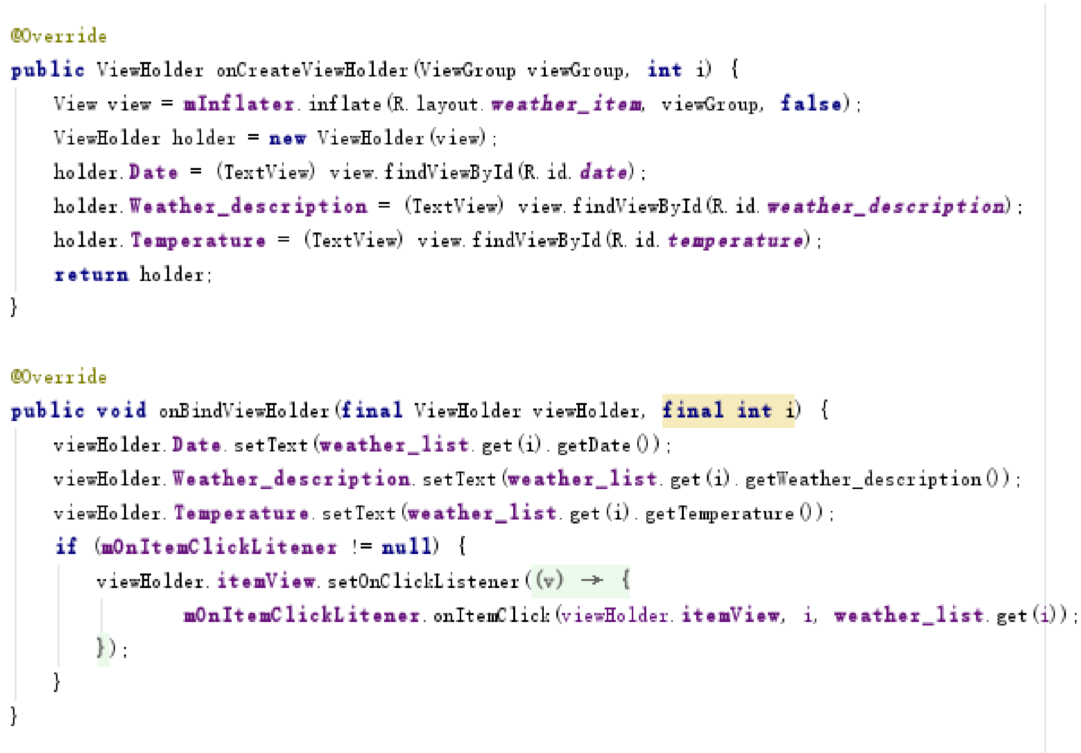


```java
public class Weather {
    private String date;
    private String weather_description;
    private String temperature;

    public String getDate() { return date; }

    public void setDate(String date) { this.date = date; }

    public String getTemperature() { return temperature; }

    public void setTemperature(String temperature) { this.temperature = temperature; }

    public String getWeather_description() { return weather_description; }

    public void setWeather_description(String weacher_description) {
        this.weather_description = weacher_description;
    }
}
```

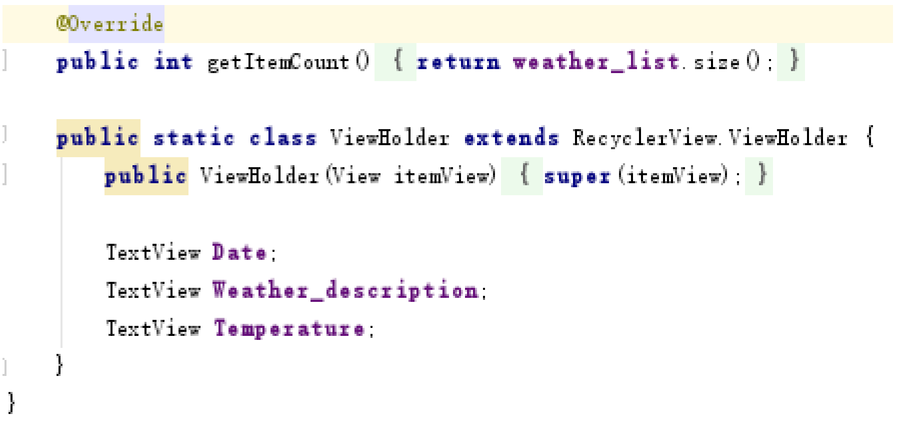
6. 构造 WeatherAdapter



```java
public class WeatherAdapter extends RecyclerView.Adapter<WeatherAdapter.ViewHolder> {
    private List<Weather> weather_list;
    private LayoutInflater mInflater;

    public interface OnItemClickLitener {
        void onItemClick(View view, int position, Weather item);
    }

    private OnItemClickLitener mOnItemClickLitener;

    public void setOnItemClickLitener(OnItemClickLitener mOnItemClickLitener) {
        this.mOnItemClickLitener = mOnItemClickLitener;
    }

    public WeatherAdapter(Context context, List<Weather> items) {
        super();
        weather_list = items;
        mInflater = LayoutInflater.from(context);
    }
```



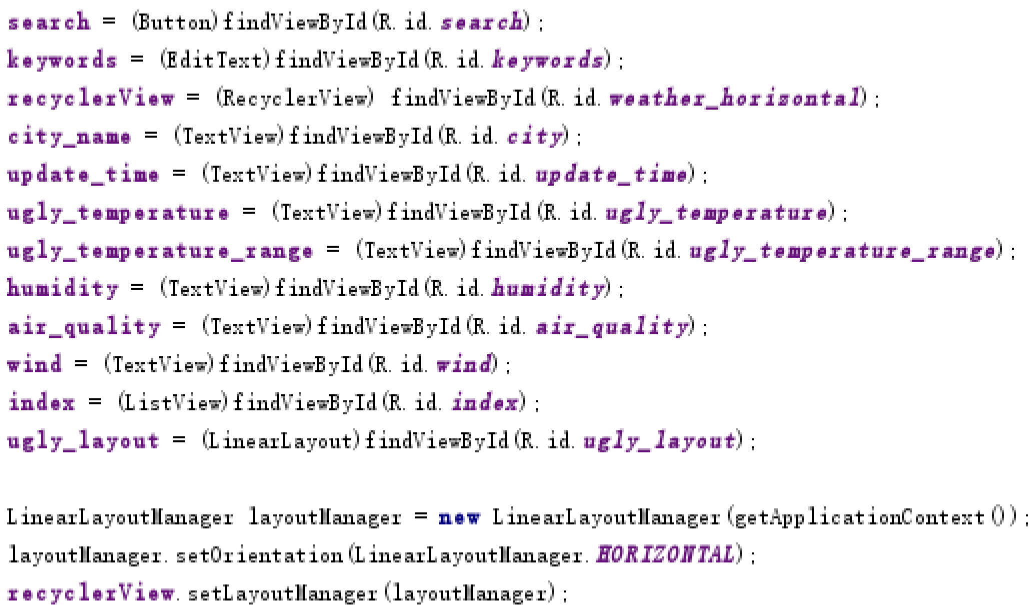
```java
@Override
public ViewHolder onCreateViewHolder(ViewGroup viewGroup, int i) {
    View view = mInflater.inflate(R.layout.weather_item, viewGroup, false);
    ViewHolder holder = new ViewHolder(view);
    holder.Date = (TextView) view.findViewById(R.id.date);
    holder.Weather_description = (TextView) view.findViewById(R.id.weather_description);
    holder.Temperature = (TextView) view.findViewById(R.id.temperature);
    return holder;
}

@Override
public void onBindViewHolder(final ViewHolder viewHolder, final int i) {
    viewHolder.Date.setText(weather_list.get(i).getDate());
    viewHolder.Weather_description.setText(weather_list.get(i).getWeather_description());
    viewHolder.Temperature.setText(weather_list.get(i).getTemperature());
    if (mOnItemClickLitener != null) {
        viewHolder.itemView.setOnClickListener((v) -> {
                mOnItemClickLitener.onItemClick(viewHolder.itemView, i, weather_list.get(i));
        });
    }
}
```



```
    @Override
    public int getItemCount() { return weather_list.size(); }

    public static class ViewHolder extends RecyclerView.ViewHolder {
        public ViewHolder(View itemView) { super(itemView); }

        TextView Date;
        TextView Weather_description;
        TextView Temperature;
    }
}
```

7. 实现 MainActivity

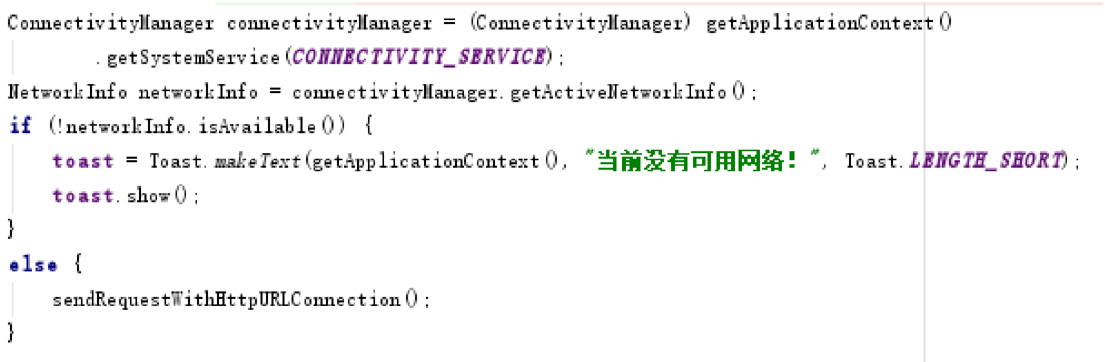
（1）获取所有的控件



```
search = (Button)findViewById(R.id.search);
keywords = (EditText)findViewById(R.id.keywords);
recyclerView = (RecyclerView) findViewById(R.id.weather_horizontal);
city_name = (TextView)findViewById(R.id.city);
update_time = (TextView)findViewById(R.id.update_time);
ugly_temperature = (TextView)findViewById(R.id.ugly_temperature);
ugly_temperature_range = (TextView)findViewById(R.id.ugly_temperature_range);
humidity = (TextView)findViewById(R.id.humidity);
air_quality = (TextView)findViewById(R.id.air_quality);
wind = (TextView)findViewById(R.id.wind);
index = (ListView)findViewById(R.id.index);
ugly_layout = (LinearLayout)findViewById(R.id.ugly_layout);

LinearLayoutManager layoutManager = new LinearLayoutManager(getApplicationContext());
layoutManager.setOrientation(LinearLayoutManager.HORIZONTAL);
recyclerView.setLayoutManager(layoutManager);
```

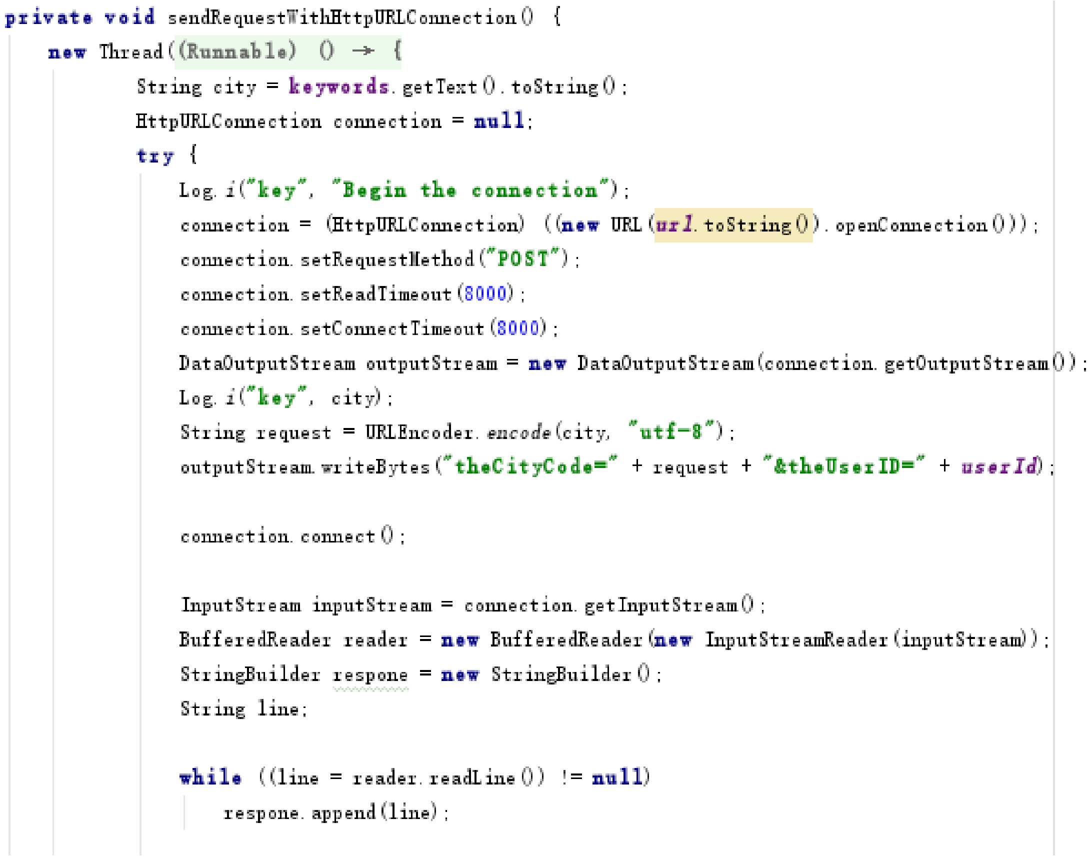
（2）实现点击 Search 的事件的回调函数

① 判断网络连接是否可用



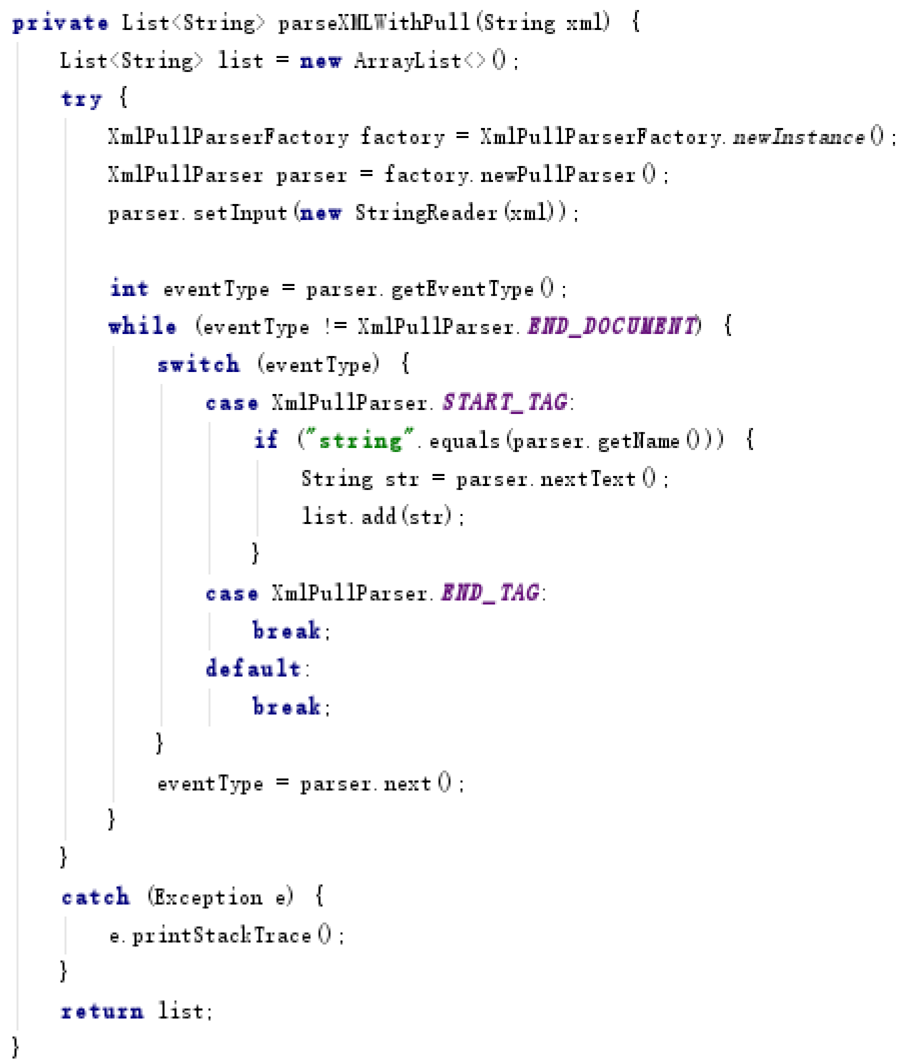
```
ConnectivityManager connectivityManager = (ConnectivityManager) getApplicationContext()
        .getSystemService(CONNECTIVITY_SERVICE);
NetworkInfo networkInfo = connectivityManager.getActiveNetworkInfo();
if (!networkInfo.isAvailable()) {
    toast = Toast.makeText(getApplicationContext(), "当前没有可用网络！", Toast.LENGTH_SHORT);
    toast.show();
}
else {
    sendRequestWithHttpURLConnection();
}
```

② 实现使用线程发送 HTTP 请求，并将相应保存到 response 中



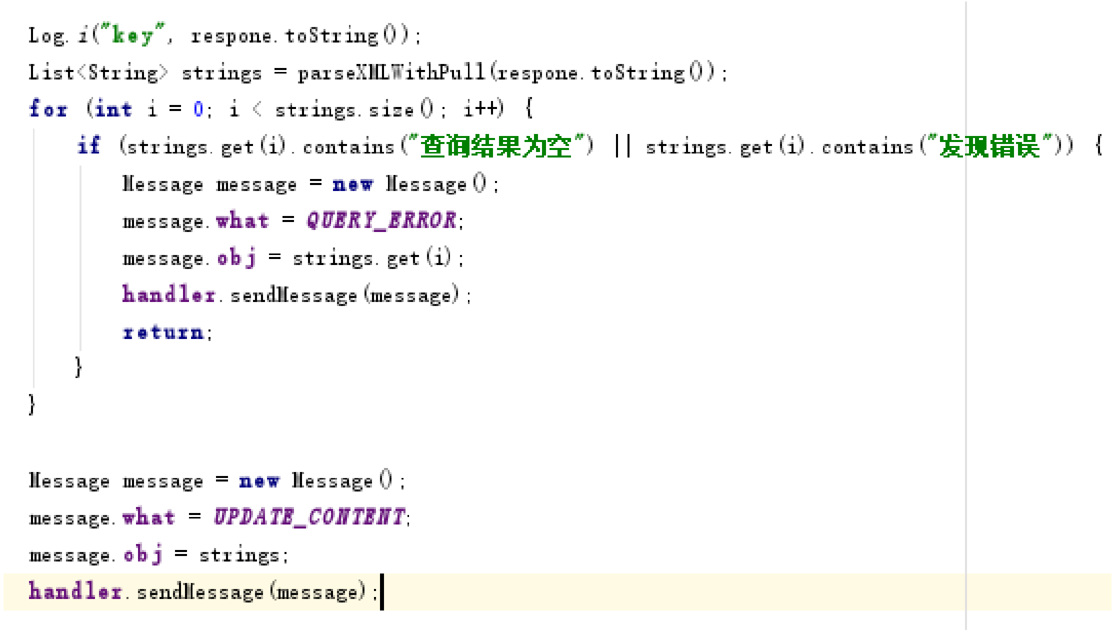
```java
private void sendRequestWithHttpURLConnection() {
    new Thread((Runnable) () -> {
            String city = keywords.getText().toString();
            HttpURLConnection connection = null;
            try {
                Log.i("key", "Begin the connection");
                connection = (HttpURLConnection) ((new URL(url.toString()).openConnection()));
                connection.setRequestMethod("POST");
                connection.setReadTimeout(8000);
                connection.setConnectTimeout(8000);
                DataOutputStream outputStream = new DataOutputStream(connection.getOutputStream());
                Log.i("key", city);
                String request = URLEncoder.encode(city, "utf-8");
                outputStream.writeBytes("theCityCode=" + request + "&theUserID=" + userId);

                connection.connect();

                InputStream inputStream = connection.getInputStream();
                BufferedReader reader = new BufferedReader(new InputStreamReader(inputStream));
                StringBuilder respone = new StringBuilder();
                String line;

                while ((line = reader.readLine()) != null)
                    respone.append(line);
```

③ 实现解析 response 中的 XML 文档



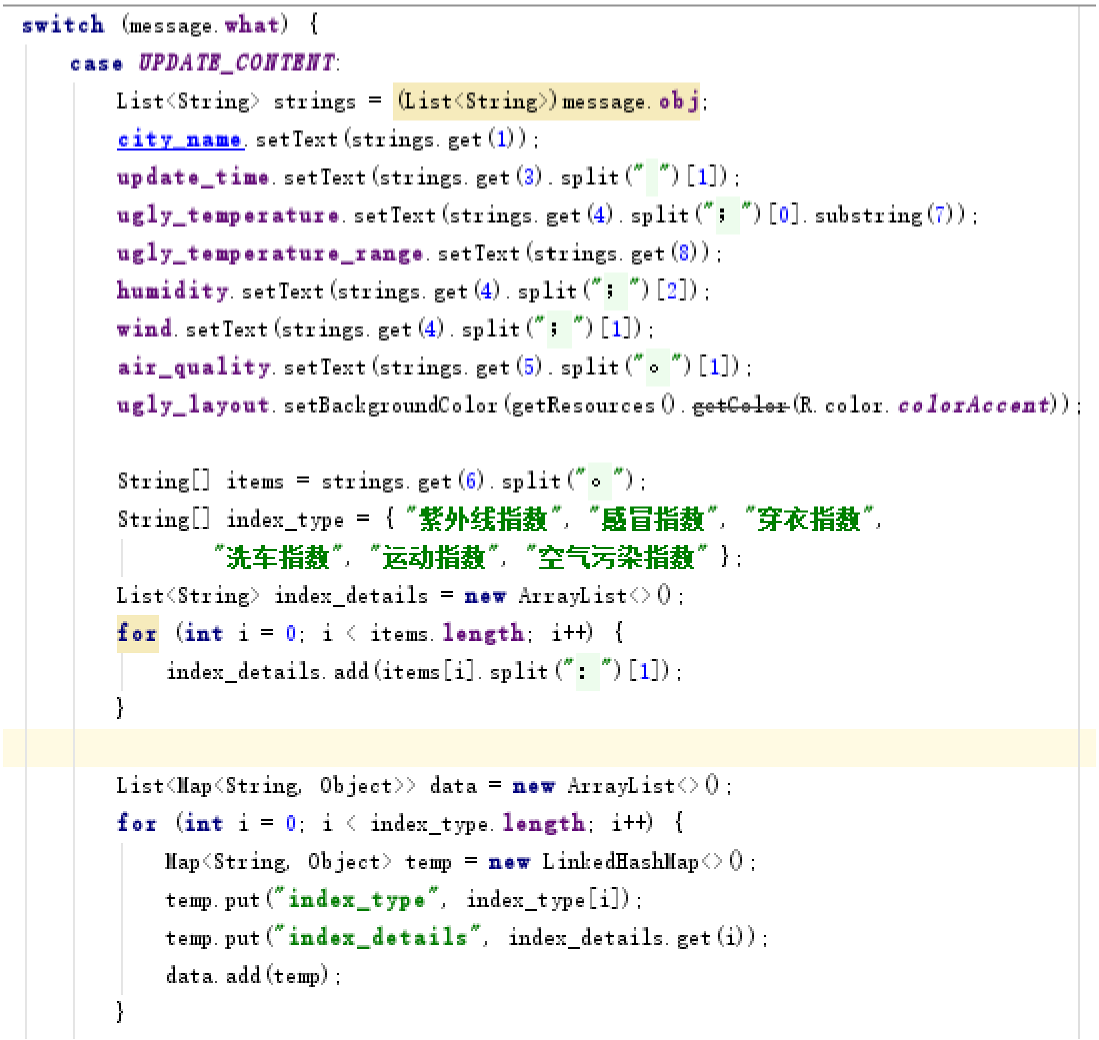
```
private List<String> parseXMLWithPull(String xml) {
    List<String> list = new ArrayList<>();
    try {
        XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
        XmlPullParser parser = factory.newPullParser();
        parser.setInput(new StringReader(xml));

        int eventType = parser.getEventType();
        while (eventType != XmlPullParser.END_DOCUMENT) {
            switch (eventType) {
                case XmlPullParser.START_TAG:
                    if ("string".equals(parser.getName())) {
                        String str = parser.nextText();
                        list.add(str);
                    }
                case XmlPullParser.END_TAG:
                    break;
                default:
                    break;
            }
            eventType = parser.next();
        }
    }
    catch (Exception e) {
        e.printStackTrace();
    }
    return list;
}
```

④ 根据返回结果的不同向 handler 发送不同的 message 类型

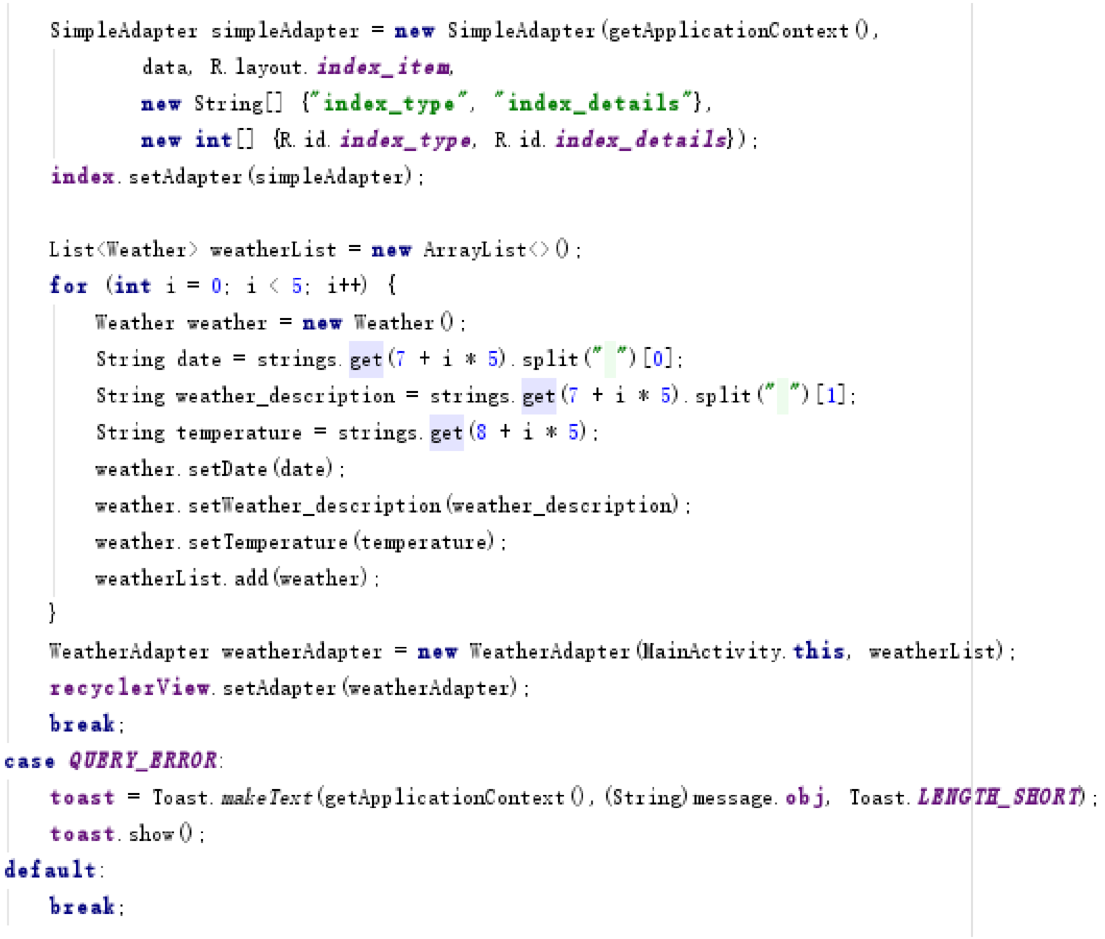


```
Log.i("key", respone.toString());
List<String> strings = parseXMLWithPull(respone.toString());
for (int i = 0; i < strings.size(); i++) {
    if (strings.get(i).contains("查询结果为空") || strings.get(i).contains("发现错误")) {
        Message message = new Message();
        message.what = QUERY_ERROR;
        message.obj = strings.get(i);
        handler.sendMessage(message);
        return;
    }
}

Message message = new Message();
message.what = UPDATE_CONTENT;
message.obj = strings;
handler.sendMessage(message);
```

⑤ 在 handler 中根据消息类型的不同更新 UI



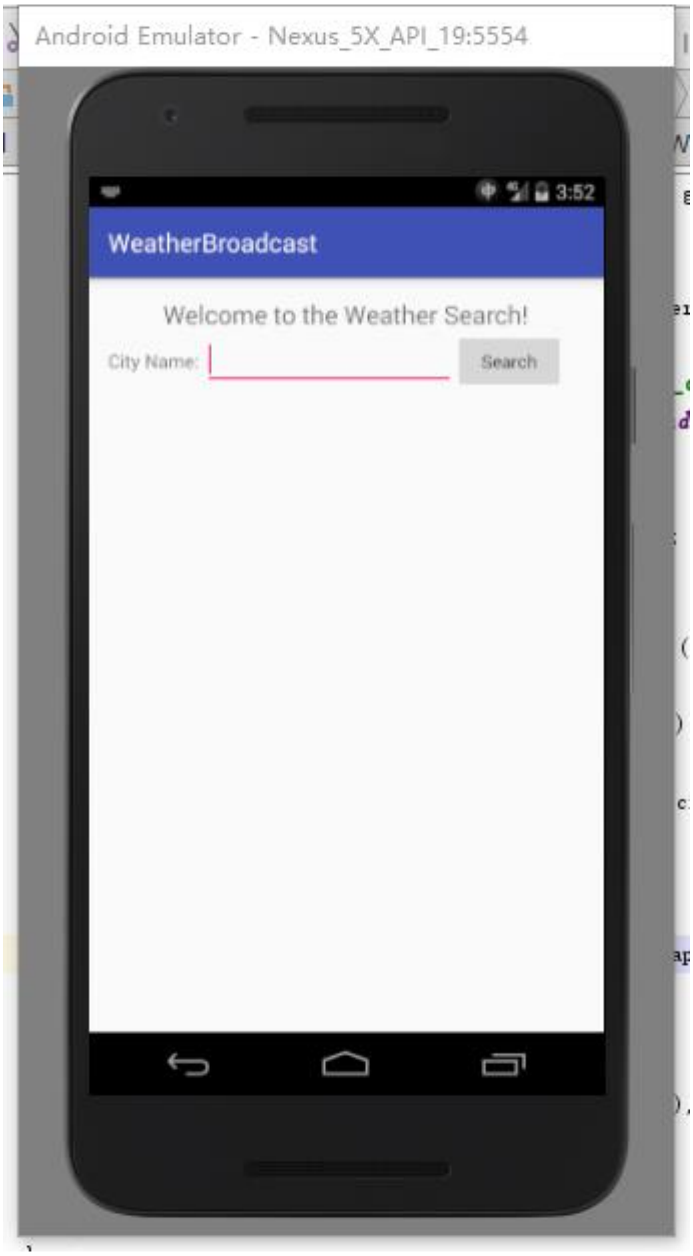
```java
switch (message.what) {
    case UPDATE_CONTENT:
        List<String> strings = (List<String>)message.obj;
        city_name.setText(strings.get(1));
        update_time.setText(strings.get(3).split("　")[1]);
        ugly_temperature.setText(strings.get(4).split("；")[0].substring(7));
        ugly_temperature_range.setText(strings.get(8));
        humidity.setText(strings.get(4).split("；")[2]);
        wind.setText(strings.get(4).split("；")[1]);
        air_quality.setText(strings.get(5).split("。")[1]);
        ugly_layout.setBackgroundColor(getResources().~~getColor~~(R.color.colorAccent));

        String[] items = strings.get(6).split("。");
        String[] index_type = { "紫外线指数", "感冒指数", "穿衣指数",
                "洗车指数", "运动指数", "空气污染指数" };
        List<String> index_details = new ArrayList<>();
        for (int i = 0; i < items.length; i++) {
            index_details.add(items[i].split("：")[1]);
        }

        List<Map<String, Object>> data = new ArrayList<>();
        for (int i = 0; i < index_type.length; i++) {
            Map<String, Object> temp = new LinkedHashMap<>();
            temp.put("index_type", index_type[i]);
            temp.put("index_details", index_details.get(i));
            data.add(temp);
        }
```



```
    SimpleAdapter simpleAdapter = new SimpleAdapter(getApplicationContext(),
            data, R.layout.index_item,
            new String[] {"index_type", "index_details"},
            new int[] {R.id.index_type, R.id.index_details});
    index.setAdapter(simpleAdapter);

    List<Weather> weatherList = new ArrayList<>();
    for (int i = 0; i < 5; i++) {
        Weather weather = new Weather();
        String date = strings.get(7 + i * 5).split(" ")[0];
        String weather_description = strings.get(7 + i * 5).split(" ")[1];
        String temperature = strings.get(8 + i * 5);
        weather.setDate(date);
        weather.setWeather_description(weather_description);
        weather.setTemperature(temperature);
        weatherList.add(weather);
    }
    WeatherAdapter weatherAdapter = new WeatherAdapter(MainActivity.this, weatherList);
    recyclerView.setAdapter(weatherAdapter);
    break;
case QUERY_ERROR:
    toast = Toast.makeText(getApplicationContext(), (String)message.obj, Toast.LENGTH_SHORT);
    toast.show();
default:
    break;
```

三、实验结果

1. 运行程序



Android Emulator - Nexus_5X_API_19:5554

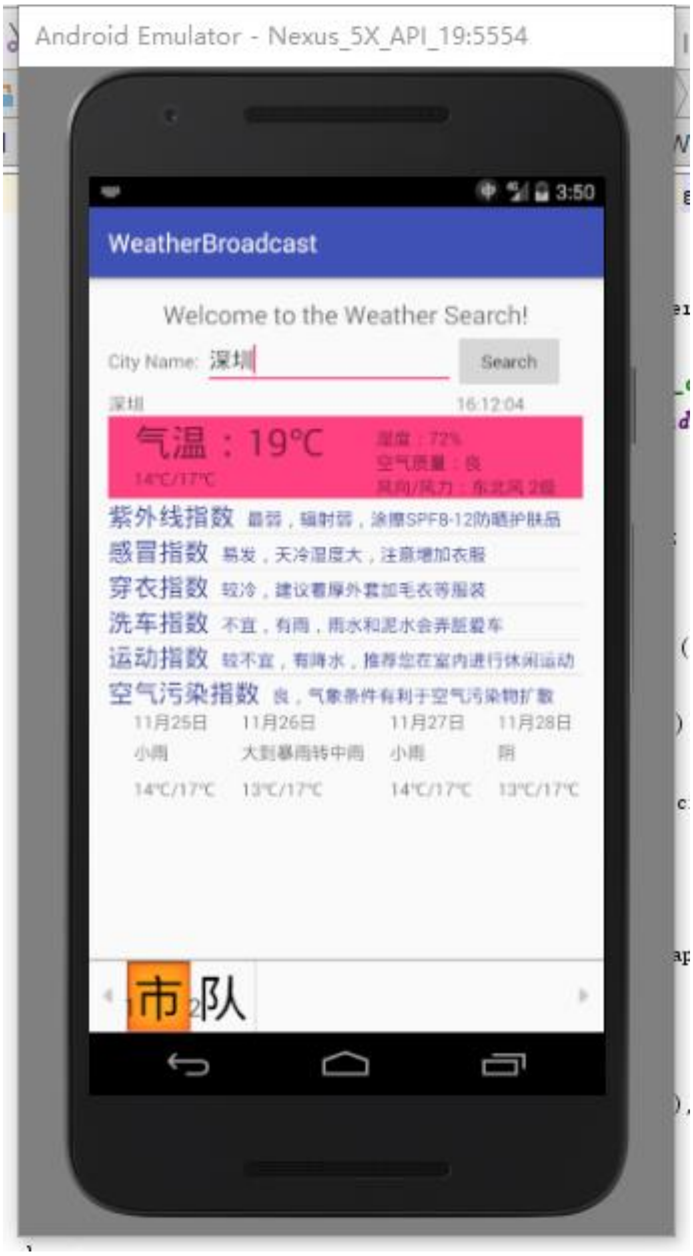
3:52

WeatherBroadcast

Welcome to the Weather Search!

City Name:

2. 查询深圳的天气



Android Emulator - Nexus_5X_API_19:5554

3:50

WeatherBroadcast

Welcome to the Weather Search!

City Name: 深圳 Search

深圳 16:12:04

气温：19℃

14℃/17℃

湿度：72%

空气质量：良

风向/风力：东北风 2级

紫外线指数 最弱，辐射弱，涂擦SPF8-12防晒护肤品

感冒指数 易发，天冷湿度大，注意增加衣服

穿衣指数 较冷，建议着厚外套加毛衣等服装

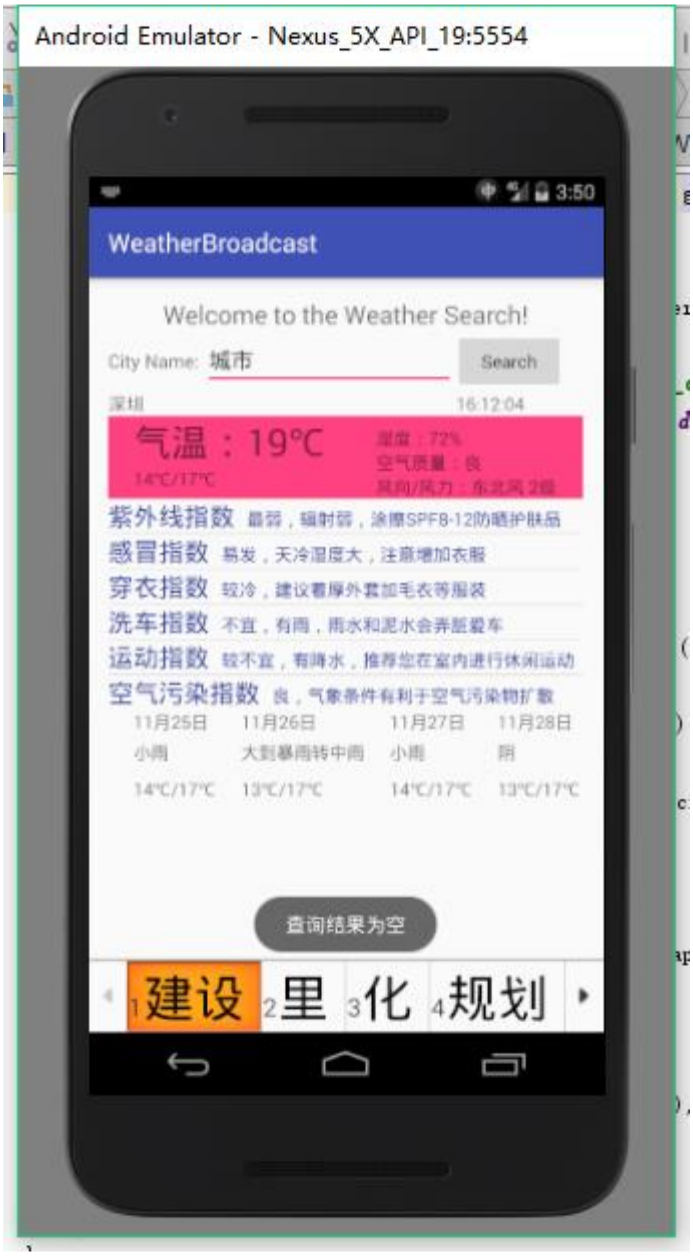
洗车指数 不宜，有雨，雨水和泥水会弄脏爱车

运动指数 较不宜，有降水，推荐您在室内进行休闲运动

空气污染指数 良，气象条件有利于空气污染物扩散

11月25日	11月26日	11月27日	11月28日
小雨	大到暴雨转中雨	小雨	阴
14℃/17℃	13℃/17℃	14℃/17℃	13℃/17℃

3. 尝试查询不存在的城市



Android Emulator - Nexus_5X_API_19:5554

3:50

WeatherBroadcast

Welcome to the Weather Search!

City Name: 城市 Search

深圳 16:12:04

气温：19℃

14℃/17℃

湿度：72%

空气质量：良

风向/风力：东北风 2级

紫外线指数 最弱，辐射弱，涂擦SPF8-12防晒护肤品

感冒指数 易发，天冷湿度大，注意增加衣服

穿衣指数 较冷，建议着厚外套加毛衣等服装

洗车指数 不宜，有雨，雨水和泥水会弄脏爱车

运动指数 较不宜，有降水，推荐您在室内进行休闲运动

空气污染指数 良，气象条件有利于空气污染物扩散

11月25日	11月26日	11月27日	11月28日
小雨	大到暴雨转中雨	小雨	阴
14℃/17℃	13℃/17℃	14℃/17℃	13℃/17℃

查询结果为空

四、实验心得

在子线程中无法显示 Toast，应该在 handler 中显示

解析 XML 文档时要很细心很细心很细心