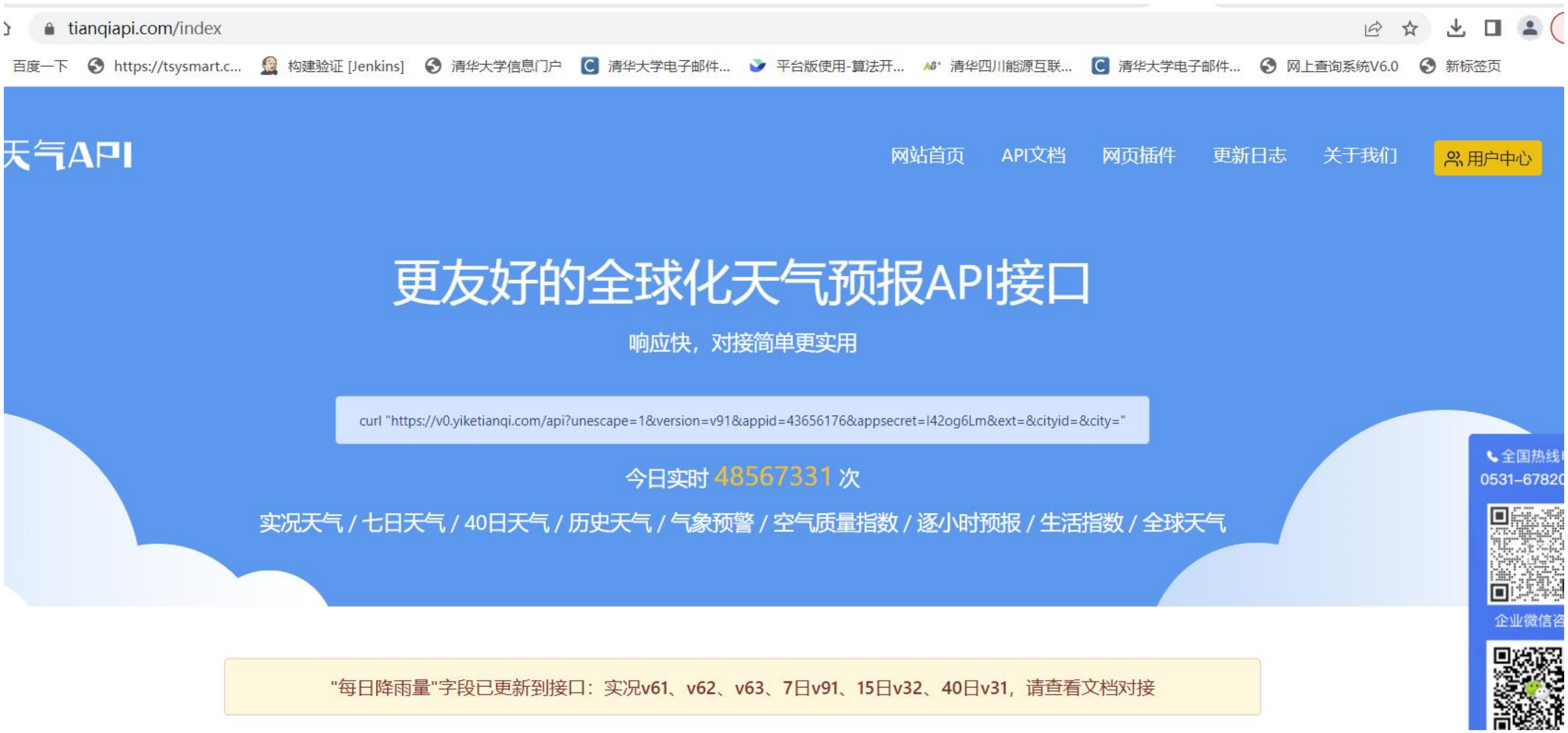
天气预报APP实战

0. 注册与登录帐号

0. 注册与登录帐号

https://tianqiapi.com/index



更友好的全球化天气预报API接口

响应快，对接简单更实用

```
curl "https://v0.yiketianqi.com/api?unescape=1&version=v91&appid=43656176&appsecret=l42og6Lm&ext=&cityid=&city="
```

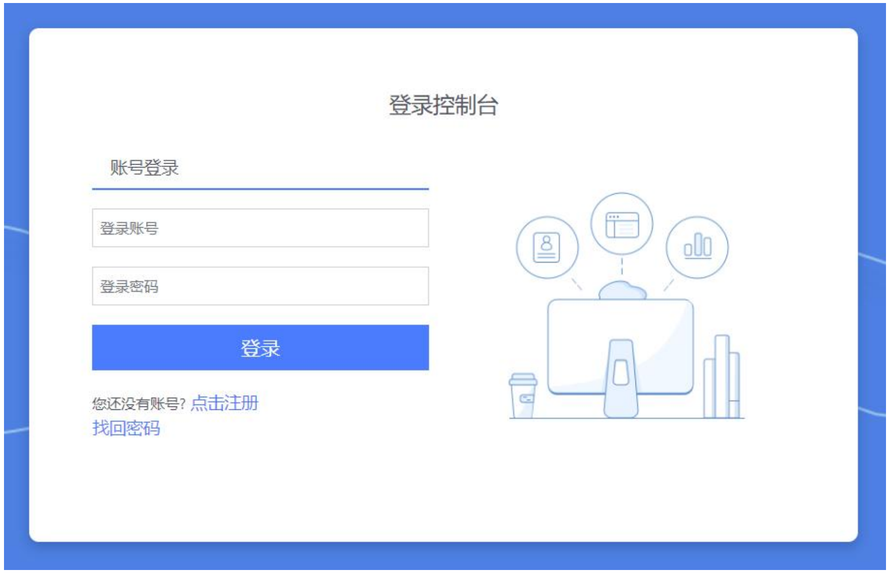
今日实时 48567331 次

实况天气 / 七日天气 / 40日天气 / 历史天气 / 气象预警 / 空气质量指数 / 逐小时预报 / 生活指数 / 全球天气

"每日降雨量"字段已更新到接口：实况v61、v62、v63、7日v91、15日v32、40日v31，请查看文档对接

0. 注册与登录帐号

https://tianqiapi.com/index



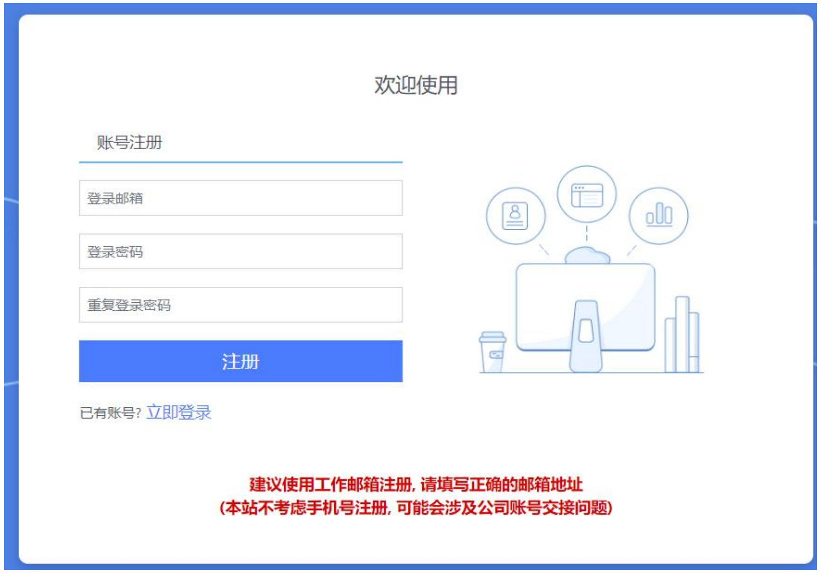
登录控制台

账号登录

登录账号

登录密码

您还没有账号? 点击注册
找回密码



欢迎使用

账号注册

登录邮箱

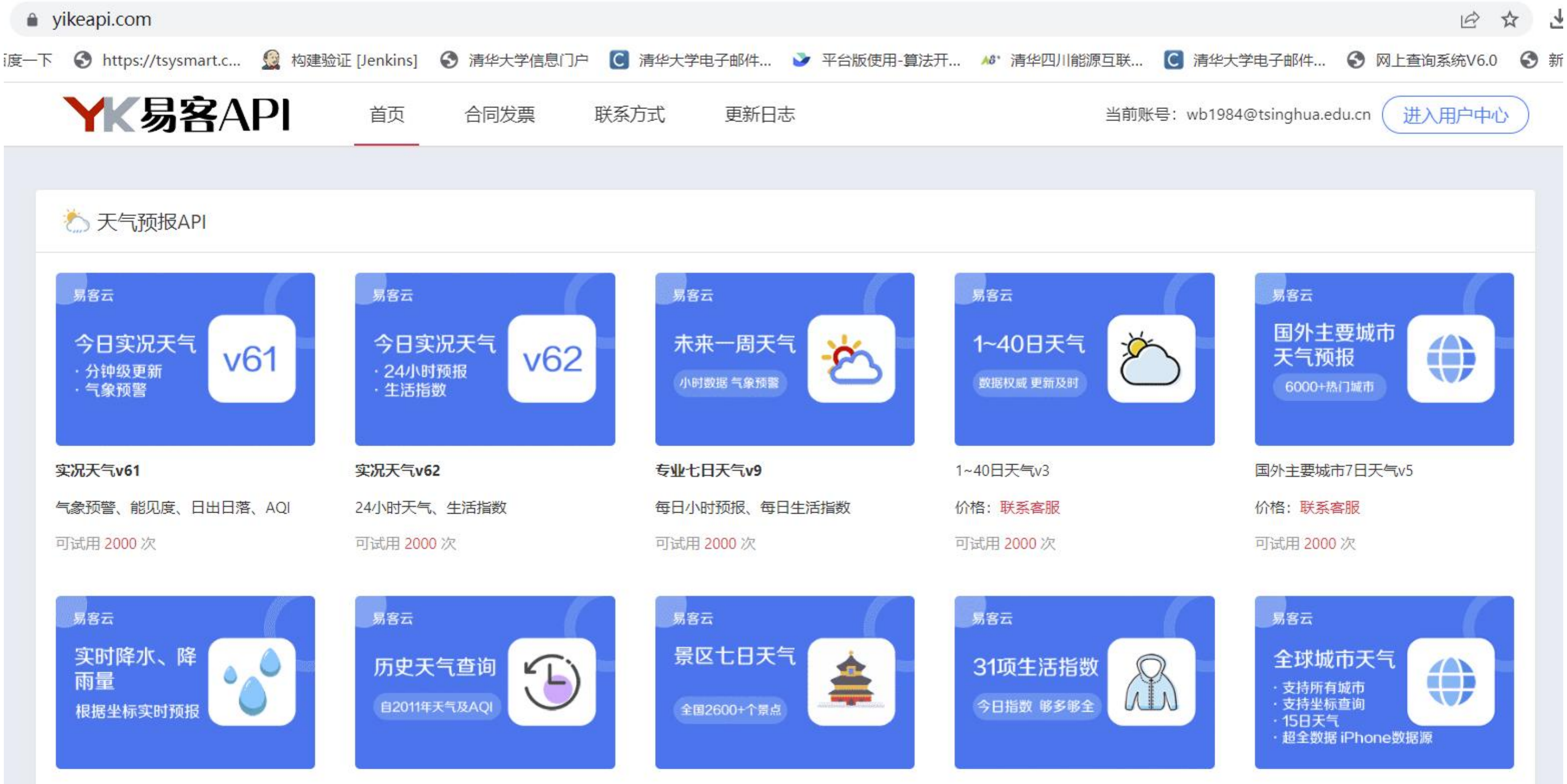
登录密码

重复登录密码

已有账号? 立即登录

建议使用工作邮箱注册，请填写正确的邮箱地址
（本站不考虑手机号注册，可能会涉及公司账号交接问题）

0. 注册与登录帐号



https://v0.yiketianqi.com/api? unescape=1 &version=v62&appid=xxx&appsecret=xxx&city=北京

https://v0.yiketianqi.com/api? unescape=1 &version=v62&appid=xxx&appsecret=xxx&cityid=101010100

v1.1 版本
获取天气预报信息

1. 获取天气预报信息

Step1:在AndroidManifest.xml中，打开查看网络状态权限。

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

1. 获取天气预报信息

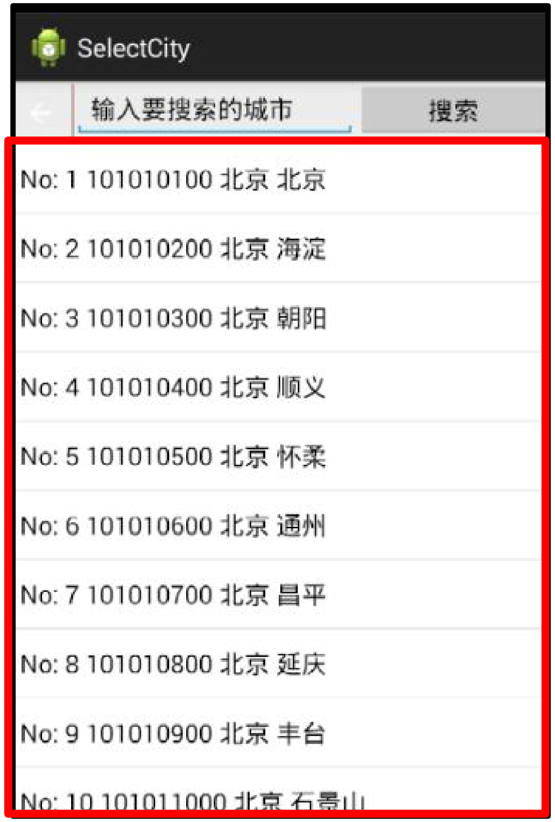
Step2:使用HttpUrlConnection,获取Url网页上的数据

```
void queryWeatherData(String cityId) {
        final String address =
        "http://v0.yiketianqi.com/api?unescape=1&version=v62&appid=xxx&appsecret=yyyy&cityid="
                + cityId;
        Log.d("Weather", address);
        TextView city_weather = (TextView) findViewById(R.id.city_weather);
        HttpURLConnection con = null;
        try {
                URL url = new URL(address);
                con = (HttpURLConnection) url.openConnection();
                con.setRequestMethod("GET");
                con.setConnectTimeout(8000);
                con.setReadTimeout(8000);
                InputStream in = con.getInputStream();
                BufferedReader reader = new BufferedReader(new InputStreamReader(in));
                StringBuilder response = new StringBuilder();
                String str;
                while ((str = reader.readLine()) != null) {
                                response.append(str);
                }
                Log.d("Weather", str);
                city_weather.setText(str);
        } catch (Exception e) {
                Log.d("Weather", e.getMessage());
                city_weather.setText(e.getMessage());
        }
}
```

v1.2 版本
增加功能：通过界面选择城市

2. 通过界面选择城市

Step1: 新建SelectCity页面，初始化listView控件



SelectCity

输入要搜索的城市

搜索

No: 1 101010100 北京 北京

No: 2 101010200 北京 海淀

No: 3 101010300 北京 朝阳

No: 4 101010400 北京 顺义

No: 5 101010500 北京 怀柔

No: 6 101010600 北京 通州

No: 7 101010700 北京 昌平

No: 8 101010800 北京 延庆

No: 9 101010900 北京 丰台

No: 10 101011000 北京 石景山

ListView控件

```
private ArrayAdapter<String> adapter;
private ArrayList<String> mArrayList;
```

在onCreate中绑定适配器

```
 mArrayList= new ArrayList<String>();
mArrayList.add("No:1 101010100  北京 北京");
mArrayList.add("No:2 101010200  北京 海淀");
mArrayList.add("No:3 101010300  北京 朝阳");
mArrayList.add("No:4 101010400  北京 顺义");
mArrayList.add("No:5 101010500  北京 怀柔");
mArrayList.add("No:6 101010600  北京 通州");
mArrayList.add("No:7 101010700  北京 昌平");
mArrayList.add("No:8 101010800  北京 延庆");
mArrayList.add("No:9 101010900  北京 丰台");
mArrayList.add("No:10 101011100  北京 石景山");
adapter = new ArrayAdapter<String>
(SelectCity.this,android.R.layout.simple_list_item_1,mArrayList);
mList.setAdapter(adapter);
```

v1.3 版本

增加功能：多线程运行

3. 在线程里获取天气预报信息

Step1:使用HttpUrlConnection,获取Url网页上的数据

```
private void queryWeatherCode(String cityName) {
      Integer cityId = mapCityInfo.get(cityName);
      final String address = "http://v0.yiketianqi.com/api?unescape=1&version=v62&appid=xxx&appsecret=yyyy&cityid="+ cityId;
      Log.d("Weather", address);
      try {
            new Thread(new Runnable() {
            public void run() {
            Log.d("Weather", "run...");
            TextView city_weather = (TextView) findViewById(R.id.city_weather);
            HttpURLConnection con = null;
            try {
                  URL url = new URL(address);
                  con = (HttpURLConnection) url.openConnection();
                  con.setRequestMethod("GET");con.setConnectTimeout(8000);con.setReadTimeout(8000);
                  InputStream in = con.getInputStream();
                  BufferedReader reader = new BufferedReader(new InputStreamReader(in));
                  StringBuilder response = new StringBuilder();
                  String str;
                  while ((str = reader.readLine()) != null) {response.append(str);}
                  Log.d("Weather", str);
                  city_weather.setText(str);
            } catch (Exception e) {
                  Log.d("Weather", e.getMessage());
            }
            }
            }).start();
      }
      catch (Exception e) {String responseStr = e.getMessage();Log.d("Weather", responseStr);}
}
```

匿名内部类实现Runnable接口

v1.4 版本
增加功能：线程安全的修改界面

4.更新界面数据

- **后台和UI分开**

✓主线程（Main Thread）

——主要负责处理与UI相关的事件，如：用户的按键事件，用户接触屏幕的事件以及屏幕绘图事 件，并把相关的事件分发到对应的组件进行处理。所以主线程通常又被叫做UI线 程。

✓子线程

——负责除UI之外较费时的操作，如从网络上下载数据，或者访问数据库

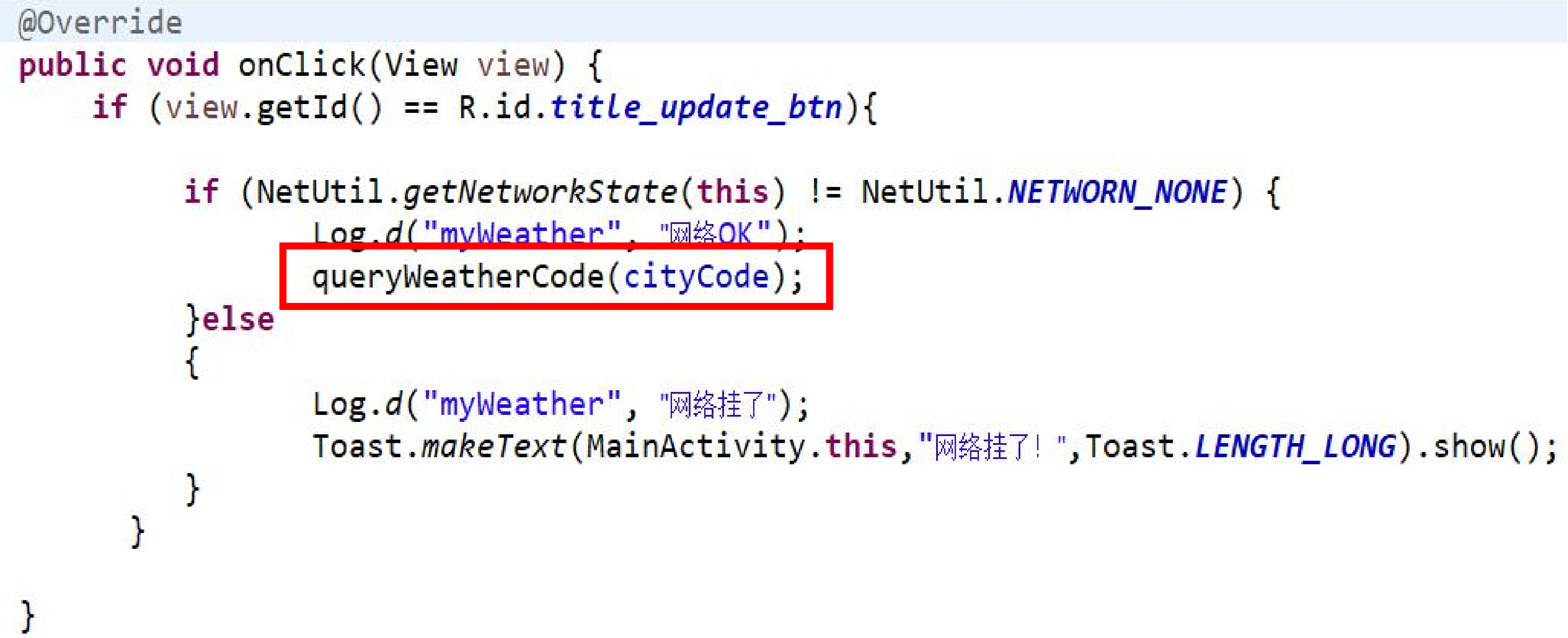
运行在主线程里的任何方法都尽可能少做事情。

可能会产生阻塞的工作不要放在主线程中。

只有UI线程才能直接操作UI控件。

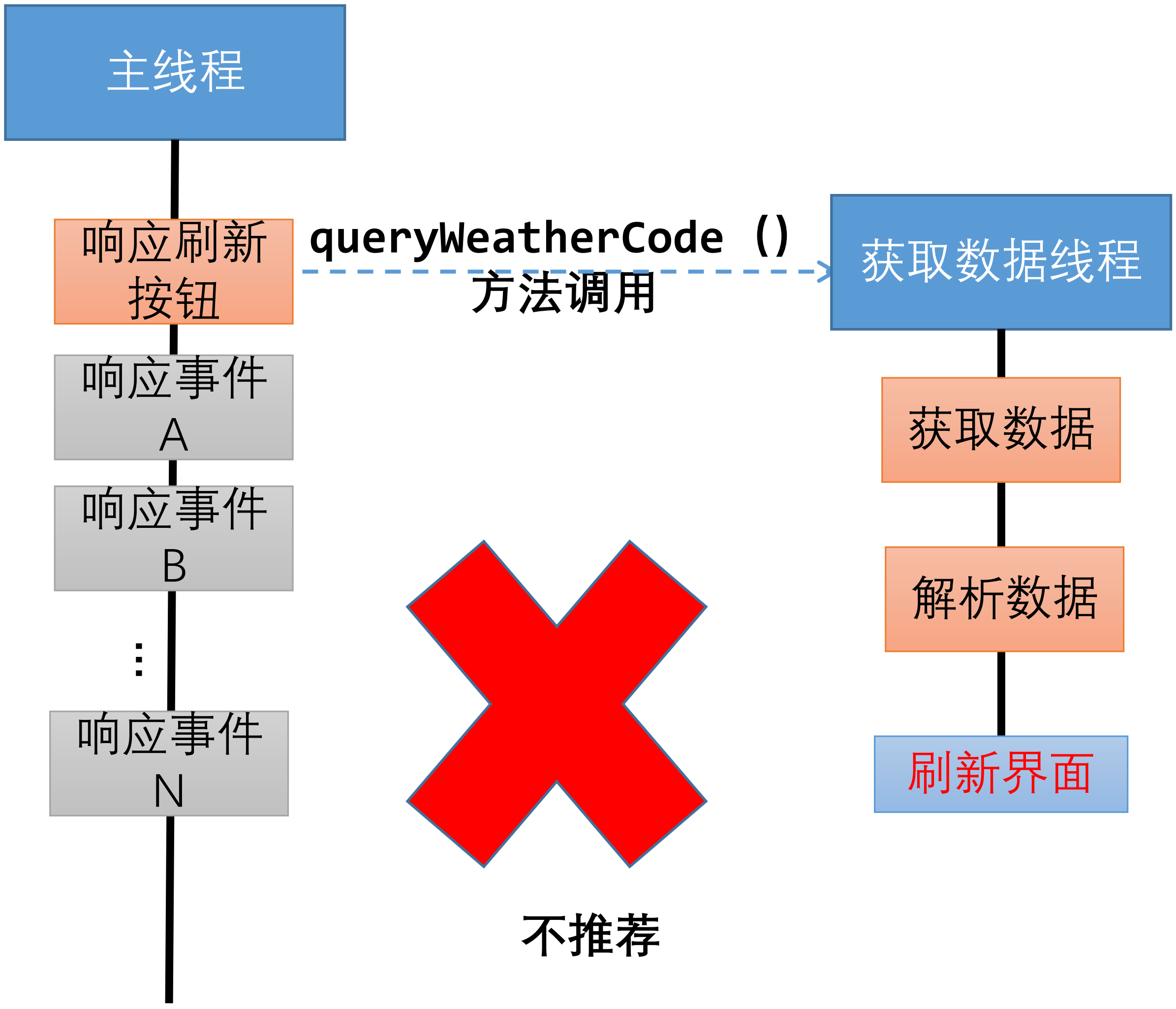
4.更新界面数据

- 主线程



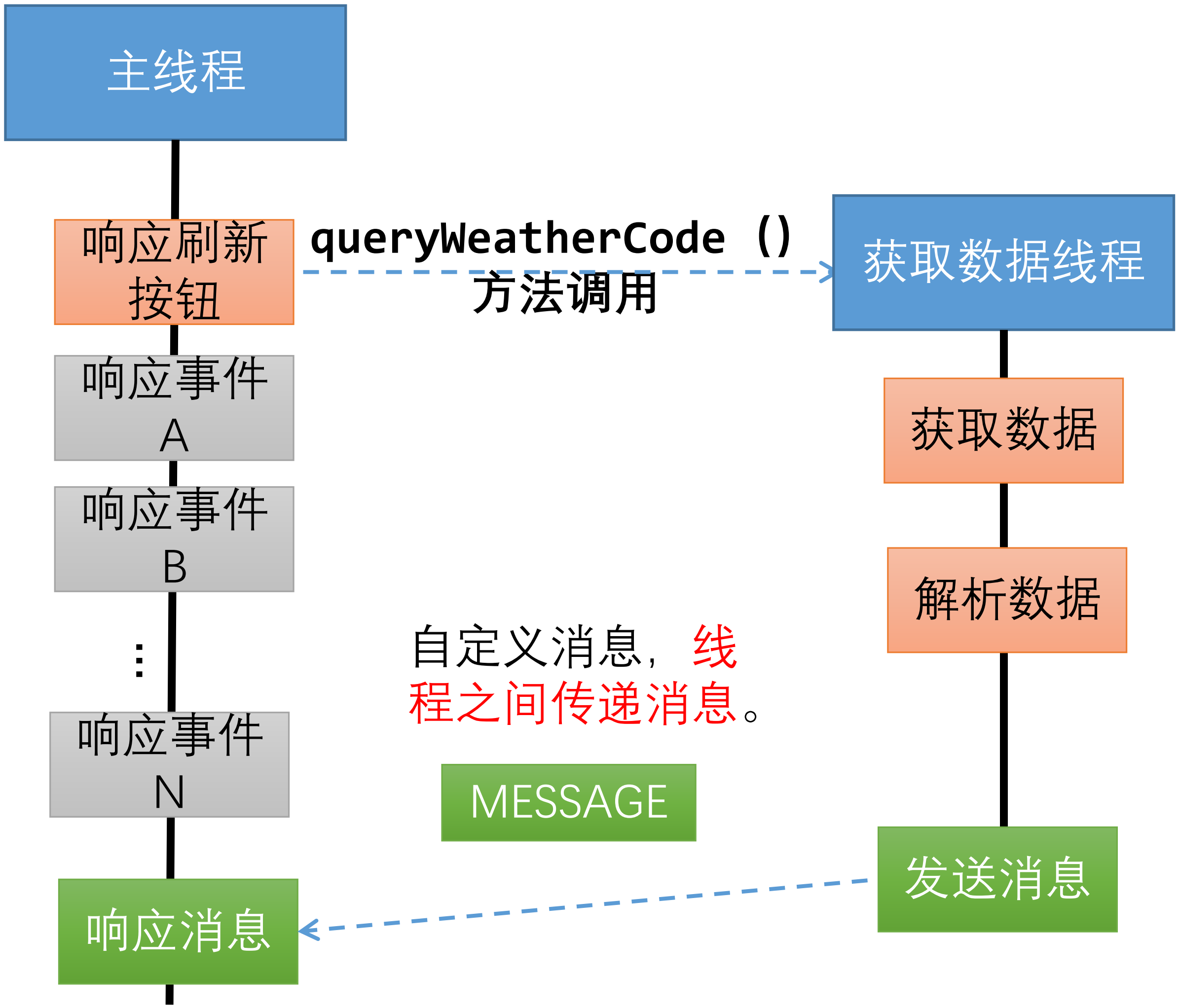
```
@Override
public void onClick(View view) {
    if (view.getId() == R.id.title_update_btn){

         if (NetUtil.getNetworkState(this) != NetUtil.NETWORN_NONE) {
                Log.d("myWeather", "网络OK");
                queryWeatherCode(cityCode);
         }else
         {
                Log.d("myWeather", "网络挂了");
                Toast.makeText(MainActivity.this,"网络挂了！",Toast.LENGTH_LONG).show();
         }
      }

}
```

4.更新界面数据


主线程
响应刷新按钮
响应事件A
响应事件B
⋮
响应事件N
queryWeatherCode（）方法调用
获取数据线程
获取数据
解析数据
刷新界面

不推荐

4.更新界面数据



主线程

响应刷新按钮

queryWeatherCode（）方法调用

获取数据线程

响应事件A

响应事件B

⋮

响应事件N

获取数据

解析数据

自定义消息，线程之间传递消息。

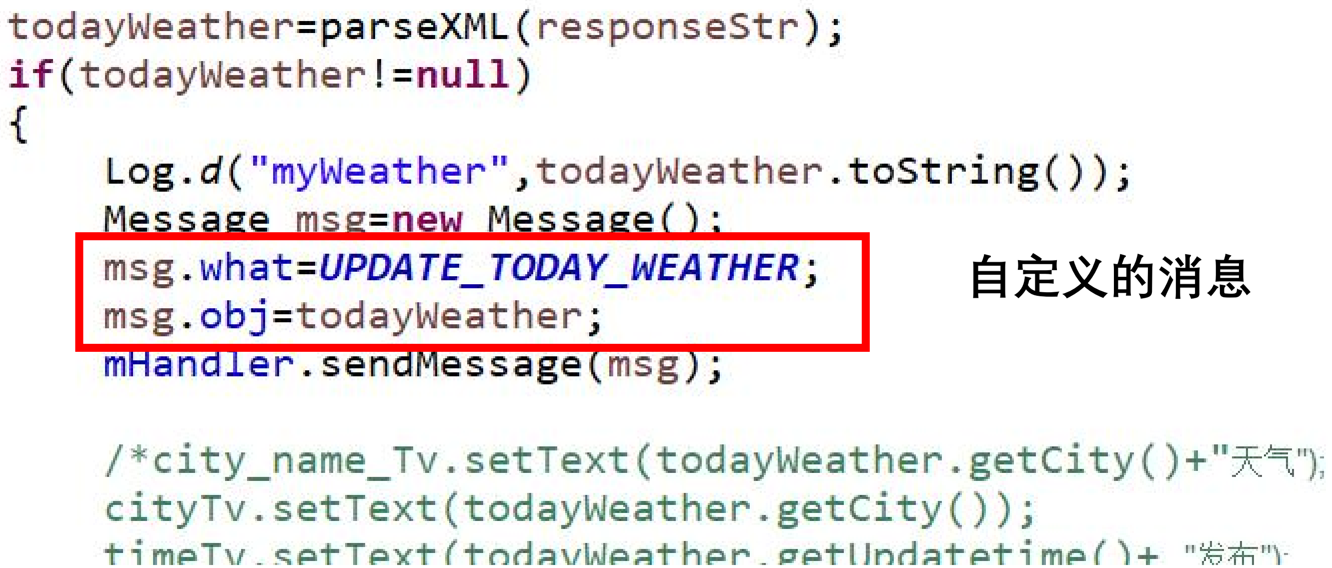
MESSAGE

发送消息

响应消息

4.更新界面数据

Step1:改造queryWeatherCode函数，发送消息

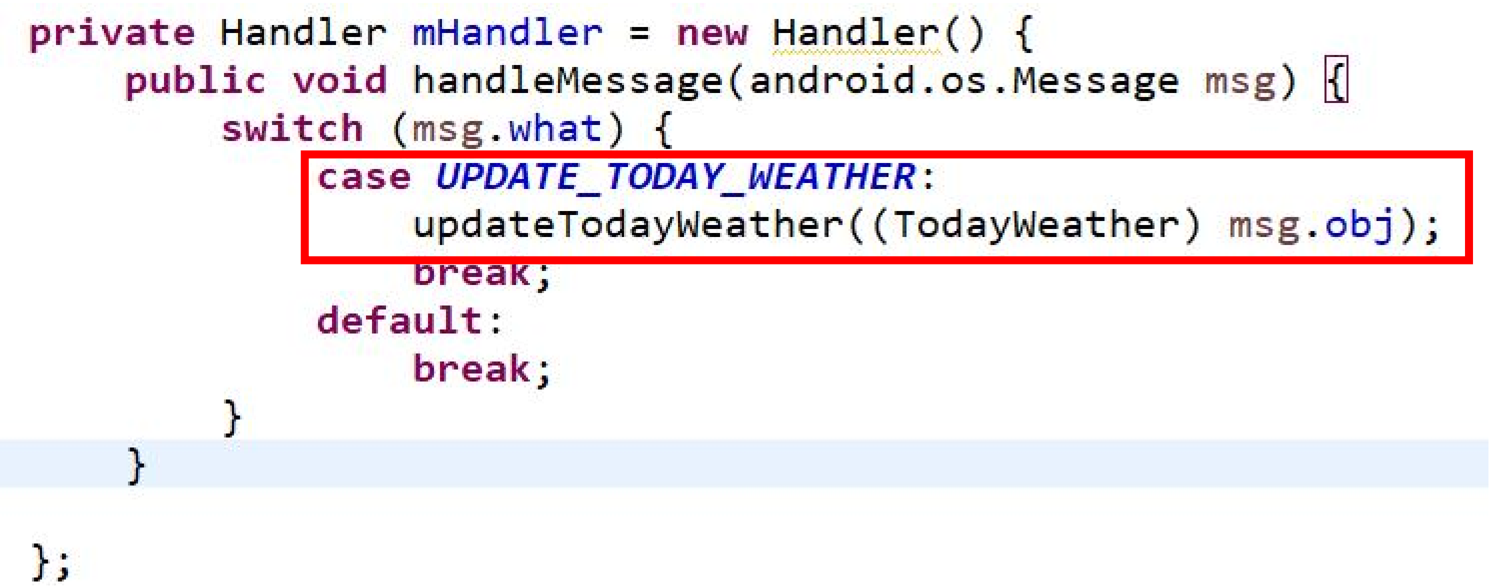


```
todayWeather=parseXML(responseStr);
if(todayWeather!=null)
{
    Log.d("myWeather",todayWeather.toString());
    Message msg=new Message();
    msg.what=UPDATE_TODAY_WEATHER;
    msg.obj=todayWeather;
    mHandler.sendMessage(msg);

    /*city_name_Tv.setText(todayWeather.getCity()+"天气");
    cityTv.setText(todayWeather.getCity());
    timeTv.setText(todayWeather.getUpdatetime()+ "发布");
```

自定义的消息

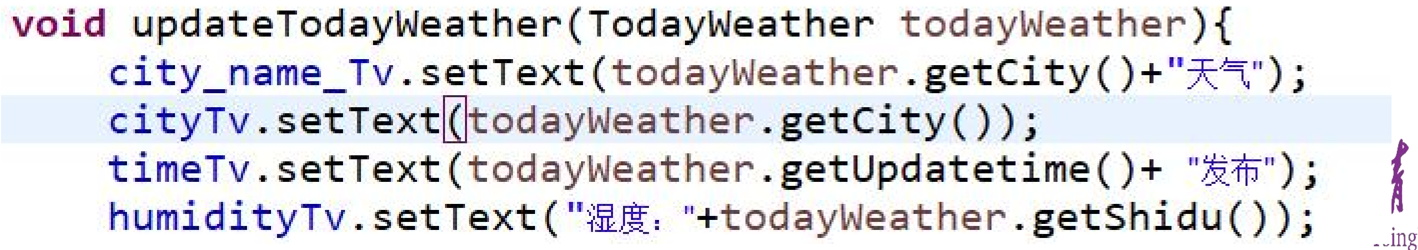
4.更新界面数据

Step2：主线程(MainActivity类)添加消息处理函数。



```
private Handler mHandler = new Handler() {
    public void handleMessage(android.os.Message msg) {
        switch (msg.what) {
            case UPDATE_TODAY_WEATHER:
                updateTodayWeather((TodayWeather) msg.obj);
                break;
            default:
                break;
        }
    }

};
```

Step3：主线程在updateTodayWeather刷新界面。



```
void updateTodayWeather(TodayWeather todayWeather){
    city_name_Tv.setText(todayWeather.getCity()+"天气");
    cityTv.setText(todayWeather.getCity());
    timeTv.setText(todayWeather.getUpdatetime()+ "发布");
    humidityTv.setText("湿度："+todayWeather.getShidu());
```

v1.5 版本

增加功能：调用json包，解析数据

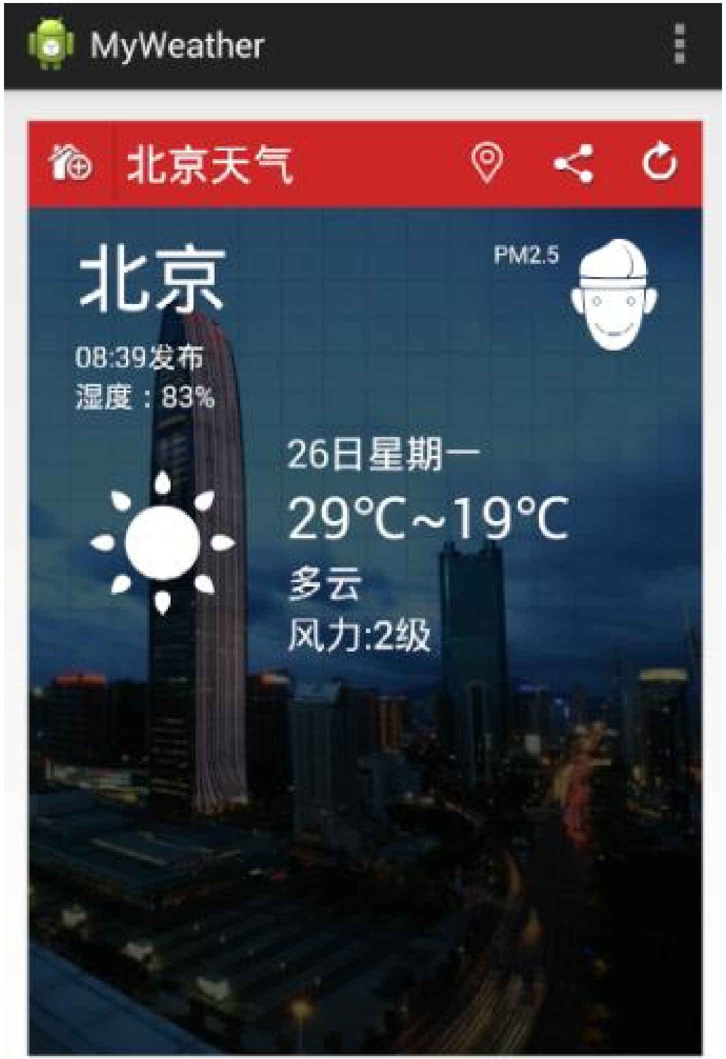
5. 解析天气预报信息

```
public void updateWeather(TodayWeather obj) {
TextView city_weather = (TextView) findViewById(R.id.city_weather);

try {
    // 定义Map集合，存储解析json字符串中各个字段的值
    Map<String, Object> map = new HashMap<String, Object>();

    // a. 构建JSONObject对象
    JSONObject jsonObject = new JSONObject(obj.toString());

    Iterator<String> keys = jsonObject.keys();

    String value = "";
    while (keys.hasNext()) {
        String key = keys.next();
        value += key + "=" + jsonObject.getString(key) + "\n";
    }
    city_weather.setText(value);
    } catch (Exception e) {
        city_weather.setText(e.getMessage());
    }
}
```

v1.6 版本

根据天气预报结果，更新画面

6. 根据天气预报结果，更新画面



MyWeather

北京天气

北京

PM2.5

08:39发布

湿度：83%

26日星期一

29℃~19℃

多云

风力:2级