Evan
Only let oneself become strong enough, good enough, can afford the life that you want to.

Android实现自动定位城市并获取天气信息

标签： android 定位 天气预报

2016-01-10 23:34 4487人阅读 评论(5) 收藏 举报

分类： 移动开发（38）

定位实现代码：

```java
<span style="font-size:14px;">import java.io.IOException;
import java.util.List;

import android.content.Context;
import android.location.Address;
import android.location.Criteria;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;

public class LocationUtils {
    public static String cityName;   //城市名
    private static Geocoder geocoder;  //此对象能通过经纬度来获取相应的城市等信息
    //通过地理坐标获取城市名 其中CN分别是city和name的首字母缩写
    public static void getCNBylocation(Context context){
        geocoder = new Geocoder(context);
        //用于获取Location对象，以及其他
        LocationManager locationManager;
        String serviceName = Context.LOCATION_SERVICE;
        //实例化一个LocationManager对象
        locationManager = (LocationManager) context.getSystemService(serviceName);
        //provider的类型
        String provider = LocationManager.NETWORK_PROVIDER;

        Criteria criteria = new Criteria();
        criteria.setAccuracy(Criteria.ACCURACY_LOW);    //低精度   高精度：ACCURACY_FINE
        criteria.setAltitudeRequired(false);       //不要求海拔
        criteria.setBearingRequired(false);       //不要求方位
        criteria.setCostAllowed(false);      //不允许产生资费
        criteria.setPowerRequirement(Criteria.POWER_LOW);   //低功耗

        //通过最后一次的地理位置来获取Location对象
        Location location = locationManager.getLastKnownLocation(provider);

        String queryed_name = updateWithNewLocation(location);
        if((queryed_name!=null)&&(0!=queryed_name.length())){
            cityName = queryed_name;
        }
        /*
        第二个参数表示更新的周期，单位为毫秒，
        第三个参数的含义表示最小距离间隔，单位是米，设定每30秒进行一次自动定位
        */
        locationManager.requestLocationUpdates(provider, 30000, 50, locationListener);
        //移除监听器，在只有一个widget的时候，这个还是适用的
        locationManager.removeUpdates(locationListener);
```

```
    }
    //方位改变是触发，进行调用
    private final static LocationListener locationListener = new LocationListener() {
        String tempCityName;
        @Override
        public void onStatusChanged(String provider, int status, Bundle extras) {
        }
        @Override
        public void onProviderEnabled(String provider) {
        }
        @Override
        public void onProviderDisabled(String provider) {
            tempCityName = updateWithNewLocation(null);
            if((tempCityName!=null)&&(tempCityName.length()!=0)){
                cityName = tempCityName;
            }
        }
        @Override
        public void onLocationChanged(Location location) {
            tempCityName = updateWithNewLocation(location);
            if((tempCityName!=null)&&(tempCityName.length()!=0)){
                cityName = tempCityName;
            }
        }
    };
    //更新location  return cityName
    private static String updateWithNewLocation(Location location){
        String mcityName = "";
        double lat = 0;
        double lng = 0;
        List<Address> addList = null;
        if(location!=null){
            lat = location.getLatitude();
            lng = location.getLongitude();
        }else{
            cityName = "无法获取地理信息";
        }
        try {
            addList = geocoder.getFromLocation(lat, lng, 1);    //解析经纬度
        } catch (IOException e) {
            e.printStackTrace();
        }
        if(addList!=null&&addList.size()>0){
            for(int i=0;i<addList.size();i++){
                Address add = addList.get(i);
                mcityName += add.getLocality();
            }
        }
        if(mcityName.length()!=0){
            return mcityName.substring(0, (mcityName.length()-1));
        }else{
            return mcityName;
        }
    }
}
</span>
```

[java]

```
<span style="font-size:14px;">public class TargetUrl {
    public final static String url1 = "http://api.map.baidu.com/telematics/v3/weather?location=";
    public final static String url2 = "&output=json&ak=9cCAXQFB468dsH11GOWL8Lx4";
}
</span>
```

根据定位到的城市名获取天气信息实现代码：

[java]

```
<span style="font-size:14px;">import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
```

```
import java.net.URL;

import org.apache.http.HttpResponse;
import org.apache.http.HttpStatus;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.util.EntityUtils;
import org.json.JSONArray;
import org.json.JSONObject;

import android.annotation.SuppressLint;
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.mine.xinlangapp.R;
import com.mine.xinlangapp.activity.MainActivity;
import com.mine.xinlangapp.location.LocationUtils;
import com.mine.xinlangapp.location.TargetUrl;

public class TupianFragment extends Fragment{
    private TextView tv, tv1, tv2, tv3, tv4, tv5;
    private ImageView iv_one, iv_two;
    private static String cityName = "";
    private String result = "";
    private static Context context = null;
    private Bitmap bitmap1, bitmap2;
    private static TupianFragment tupian = null;
    public static int tupian_hour = 60;
    private static Handler handler3 = new Handler();
    @SuppressWarnings("deprecation")
    private static Runnable runnable = new Runnable() {
        @Override
        public void run() {
            tupian.getActivity().removeDialog(0);
            Toast.makeText(tupian.getActivity(), "加载失败", Toast.LENGTH_SHORT).show();
    //      handler3.postDelayed(this, 2000);    //每两秒执行一次runnable
        }
    };
    //自动刷新
    private Runnable runnable2 = new Runnable() {
        @Override
        public void run() {
            tupian.send(cityName);
            Message m = tupian.handler.obtainMessage();
            tupian.handler.sendMessage(m);
            handler3.postDelayed(this, tupian_hour*3600*1000);
        }
    };
    @SuppressLint("HandlerLeak")
    @SuppressWarnings("deprecation")
    public static Handler handler1 = new Handler(){
        public void handleMessage(Message msg){
            tupian.getActivity().showDialog(0);
            //启动定时器
            handler3.postDelayed(runnable, 5000);   //五秒后执行
            new Thread(new Runnable() {
                @Override
                public void run() {
                    tupian.send(cityName);
                    Message m = tupian.handler.obtainMessage();
                    tupian.handler.sendMessage(m);
                }
            }).start();
        }
    };
    @SuppressLint("HandlerLeak")
```

```
84.     private Handler handler = new Handler(){
85.         public void handleMessage(Message msg){
86.             if(result != null){
87.                 try {
88.                     JSONObject datajson = new JSONObject(result);  //第一步，将String格式转换回json格式
89.                     JSONArray results = datajson.getJSONArray("results");  //获取results数组
90.
91.                     JSONObject city = results.getJSONObject(0);
92.                     String currentCity = city.getString("currentCity");  //获取city名字
93.                     String pm25 = city.getString("pm25");   //获取pm25
94.                     tv.setText("城市："+currentCity+"\n"+"pm25："+pm25);  //测试城市和pm25
95.                     JSONArray index = city.getJSONArray("index"); //获取index里面的JSONArray
96.                     //获取穿衣
97.                     JSONObject cy = index.getJSONObject(0);
98.                     String titlec = cy.getString("title");
99.                     String zsc = cy.getString("zs");
100.                    String tiptc = cy.getString("tipt");
101.                    String desc = cy.getString("des");
102.                    //获取洗车
103.                    JSONObject xc = index.getJSONObject(1);
104.                    String titlex = xc.getString("title");
105.                    String zsx = xc.getString("zs");
106.                    String tiptx = xc.getString("tipt");
107.                    String desx = xc.getString("des");
108.                    tv1.setText(titlec+" : "+zsc+"\n"+tiptc+" : "+desc);
109.                    tv2.setText(titlex+" : "+zsx+"\n"+tiptx+" : "+desx);
110.
111.                    //weather_data, 未来几天
112.                    JSONArray weather_data = city.getJSONArray("weather_data");
113.                    //获取今天
114.                    JSONObject today = weather_data.getJSONObject(0);
115.                    String date0 = today.getString("date");
116.                    final String dayPictureUrl0 = today.getString("dayPictureUrl");
117.                    final String nightPictureUrl0 = today.getString("nightPictureUrl");
118.                    String weather0 = today.getString("weather");
119.                    String wind0 = today.getString("wind");
120.                    String temperature0 = today.getString("temperature");
121.                    tv3.setText("\n"+"今天："+date0+"\n"+"实时："+weather0+"\n"+"风力："+
122.                    wind0+"\n"+"温度范围："+temperature0+"\n");
123.
124.                    //获取明天
125.                    JSONObject tomorrow = weather_data.getJSONObject(1);
126.                    String date1 = tomorrow.getString("date");
127.                    String weather1 = tomorrow.getString("weather");
128.                    String wind1 = tomorrow.getString("wind");
129.                    String temperature1 = tomorrow.getString("temperature");
130.                    tv4.setText("明天："+date1+"\n"+weather1+"\n"+
131.                    "风力："+wind1+"\n"+"温度范围："+temperature1+"\n");
132.
133.                    //获取后天
134.                    JSONObject after_tomorrow = weather_data.getJSONObject(2);
135.                    String date2 = after_tomorrow.getString("date");
136.                    String weather2 = after_tomorrow.getString("weather");
137.                    String wind2 = after_tomorrow.getString("wind");
138.                    String temperature2 = after_tomorrow.getString("temperature");
139.                    tv5.setText("后天："+date2+"\n"+weather2+"\n"+
140.                    "风力："+wind2+"\n"+"温度范围："+temperature2+"\n");
141.
142.                    new Thread(new Runnable() {
143.                        @Override
144.                        public void run() {
145.                            bitmap1 = returnBitMap(dayPictureUrl0);
146.                            bitmap2 = returnBitMap(nightPictureUrl0);
147.                            Message m = handler2.obtainMessage();
148.                            handler2.sendMessage(m);
149.                        }
150.                    }).start();
151.                } catch (Exception e) {
152.                    e.printStackTrace();
153.                }
154.            }
155.            super.handleMessage(msg);
156.        }
157.    };
158.    @SuppressWarnings("deprecation")
159.    @SuppressLint("HandlerLeak")
160.    private Handler handler2 = new Handler(){
161.        public void handleMessage(Message msg){
162.            if(bitmap1!=null)
```

```
                iv_one.setImageBitmap(bitmap1);
            if(bitmap2!=null)
                iv_two.setImageBitmap(bitmap2);
            if(bitmap1!=null&&bitmap2!=null){
                //停止计时器
                handler3.removeCallbacks(runnable);
                tupian.getActivity().removeDialog(0);
            }
        }
    };
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
            Bundle savedInstanceState){
        context = TupianFragment.this.getActivity();
        tupian = TupianFragment.this;
        LocationUtils.getCNBylocation(context);
        cityName = LocationUtils.cityName;
        MainActivity.text.setText(cityName);

        View view = inflater.inflate(R.layout.tupianfragment, container,false);
        iv_one = (ImageView) view.findViewById(R.id.iv_one);
        iv_two = (ImageView) view.findViewById(R.id.iv_two);
        tv = (TextView) view.findViewById(R.id.tv);
        tv1 = (TextView) view.findViewById(R.id.tv1);
        tv2 = (TextView) view.findViewById(R.id.tv2);
        tv3 = (TextView) view.findViewById(R.id.tv3);
        tv4 = (TextView) view.findViewById(R.id.tv4);
        tv5 = (TextView) view.findViewById(R.id.tv5);
        //启动计时器
        handler3.postDelayed(runnable2, tupian_hour*3600*1000);

        new Thread(new Runnable() {
            @Override
            public void run() {
                send(cityName);
                Message m = handler.obtainMessage();
                handler.sendMessage(m);
            }
        }).start();

        return view;
    }
    private String send(String city){
        String target = TargetUrl.url1+city+TargetUrl.url2;  //要提交的目标地址
        HttpClient httpclient = new DefaultHttpClient();
        HttpGet httpRequest = new HttpGet(target);  //创建HttpGet对象
        HttpResponse httpResponse = null;
        try {
            httpResponse = httpclient.execute(httpRequest);
            if(httpResponse.getStatusLine().getStatusCode() == HttpStatus.SC_OK){
                result = EntityUtils.toString(httpResponse.getEntity()).trim();  //获取返回的字符串
            }else{
                result = "fail";
            }
        } catch (ClientProtocolException e) {
            e.printStackTrace();
        }catch (IOException e) {
            e.printStackTrace();
        }
        return null;
    }
    //以Bitmap的方式获取一张图片
    public Bitmap returnBitMap(String url){
        URL myFileUrl = null;
        Bitmap bitmap = null;
        try{
            myFileUrl = new URL(url);
        }catch(MalformedURLException e){
            e.printStackTrace();
        }
        try{
            HttpURLConnection conn = (HttpURLConnection) myFileUrl.openConnection();
            conn.setDoInput(true);
            conn.connect();
            InputStream is = conn.getInputStream();
            bitmap = BitmapFactory.decodeStream(is);
            is.close();
        }catch(IOException e){
            e.printStackTrace();
```

```
        }
        return bitmap;
    }
    @Override
    public void onDestroy() {
        //停止计时器
        handler3.removeCallbacks(runnable2);
        super.onDestroy();
    }
}
</span>
```

最后别忘记添加权限：

```
[html]
<span style="font-size:14px;">    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" /></span>
```

说明：

```
[java]
<span style="font-size:14px;">criteria.setAccuracy(Criteria.ACCURACY_LOW);    //低精度    高精度：ACCURACY_FINE
使用网络定位要选择低精度，如果选择了高精度它会第一选择为：GPS定位，没有开启GPS定位，才会使用网络定位。
</span>
```


- 上一篇 Android之实现定位
- 下一篇 仿微信中加载网页时带线行进度条的WebView的实现

相关文章推荐

- Android获取所在地城市名
- android GPS 获取城市信息
- Android进阶之利用百度地图定位获取当前城市
- Android GPS定位，取得城市名称
- Android 根据IP地址获取城市
- Android——使用百度API获取经度纬度以及所在...
- Android获取当前的城市名的方法
- Android百度地图API获取当前位置和当前城市
- Android 实现省份城市的选择，并获取城市编号
- 安卓盒子launcher界面开发之添加自动定位，获取...