



后台自动更新天气

加入后台自动更新天气的功能，这样可以尽可能地保证用户每次打开软件时看到的都是最新的天气信息。

1. 创建自动更新 Service

新建 service 包，创建 AutoUpdateService 类，并将 Exported 和 Enabled 这两个属性勾选，代码如下所示：

```
package com.coolweather.android.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 android.util.Log;
import android.widget.Toast;
import com.bumptech.glide.Glide;
import com.coolweather.android.R;
import com.coolweather.android.WeatherActivity;
import com.coolweather.android.gson.Weather;
import com.coolweather.android.util.HttpUtil;
import com.coolweather.android.util.Utility;
import java.io.IOException;
import okhttp3.Call;
import okhttp3.Callback;
import okhttp3.Response;
```



```
public class AutoUpdateService extends Service {
    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }
    @Override
    public int onStartCommand(Intent intent, int flags, int startId)
    {
        updateWeather();
        updateBingPic();
        AlarmManager manager = (AlarmManager)
getSystemService(ALARM_SERVICE);
        int anHour = 8 * 60 * 60 * 1000; // 这是8小时的毫秒数
        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);
    }
    /**
     * 更新天气信息。
     */
    private void updateWeather() {
        SharedPreferences prefs =
PreferenceManager.getDefaultSharedPreferences(this);
        String weatherString = prefs.getString("weather", null);
        if (weatherString != null) {
```



```
// 有缓存时直接解析天气数据
Weather weather =
Utility.handleWeatherResponse(weatherString);
String weatherId = weather.basic.weatherId;
String weatherUrl =
"http://guolin.tech/api/weather?cityid=" + weatherId +
"&key=bc0418b57b2d4918819d3974ac1285d9";
HttpUtil.sendOkHttpRequest(weatherUrl, new Callback() {
    @Override
    public void onResponse(Call call, Response response)
throws IOException {
        String responseText = response.body().string();
        Weather weather =
Utility.handleWeatherResponse(responseText);
        if (weather != null && "ok".equals(weather.status))
        {
            SharedPreferences.Editor editor =
PreferenceManager.getDefaultSharedPreferences(AutoUpdateService.this)
.edit();
            editor.putString("weather", responseText);
            editor.apply();
        }
    }
    @Override
    public void onFailure(Call call, IOException e) {
        e.printStackTrace();
    }
});
}
```



```
    }  
    /**  
     * 更新必应每日一图  
     */  
    private void updateBingPic() {  
        String requestBingPic = "http://guolin.tech/api/bing_pic";  
        HttpUtil.sendOkHttpRequest(requestBingPic, new Callback() {  
            @Override  
            public void onResponse(Call call, Response response)  
                throws IOException {  
                String bingPic = response.body().string();  
                SharedPreferences.Editor editor =  
                    PreferenceManager.getDefaultSharedPreferences(AutoUpdateService.this)  
                        .edit();  
                editor.putString("bing_pic", bingPic);  
                editor.apply();  
            }  
            @Override  
            public void onFailure(Call call, IOException e) {  
                e.printStackTrace();  
            }  
        });  
    }  
}
```

在 `onStartCommand()` 方法中先调用 `updateWeather()` 方法更新天气，然后调用 `updateBingPic()` 方法来更新背景图片，我们将更新后的数据直接存储到 `SharedPreferences` 文件中。因为打开 `WeatherActivity` 的时候都会优先从 `SharedPreferences` 缓存中读取数据。



之后就是创建定时任务，我们设置时间间隔是 8 小时，8 小时后 AutoUpdateReceiver 的 onStartCommand () 方法就会重新执行，这样也就实现后台定时更新了。

另外需要在 WeatherActivity 中激活 AutoUpdateService 服务。

```
/**
 * 处理并展示 Weather 实体类中的数据。
 */
private void showWeatherInfo(Weather weather) {
    .....
    Intent intent = new Intent(this, AutoUpdateService.class);
    startService(intent);
}
```

在 WeatherActivity 最后加入启动 AutoUpdateService 这个服务代码，这样一旦选中某个城市并成功更新天气之后，AutoUpdateService 就一直在后台运行并保证 8 小时更新一次天气。

2. 修改图标和名称

按照工程要求，应该提供几种不同分辨率的图片，然后分别放入到相应分辨率的 mipmap 目录下。本程序只使用一张图片，名称为：logo.png，放入到所有已 mipmap 开头的目录下，然后修改 AndroidManifest.xml 中的代码，如下所示：

```
<application
    android:name="org.litepal.LitePalApplication"
    android:allowBackup="true"
    android:icon="@mipmap/logo"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    .....

```

然后在修改程序名称，打开 res/values/string.xml，找到 app_name，将它修改成酷欧天气即可。