

SimpleWeather

A SAMPLE 2019-6-1

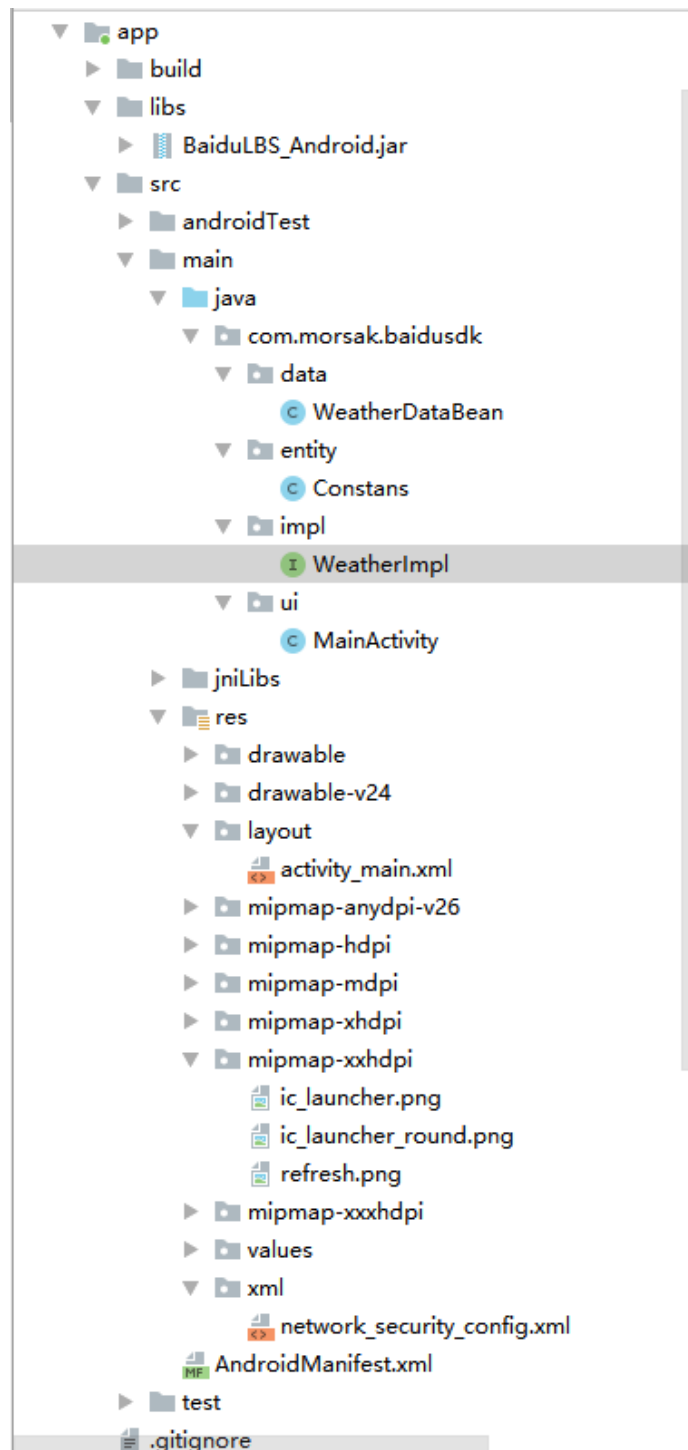
MORSAK

Github 项目分享: <https://github.com/MorSaK/BaiduSDK>

涉及技术: BaiduSDK Retrofit2.5 Gson2.5 NetworkSecurity

采用语言: JAVA

代码框架:



首先, 按照我的习惯, 先随便搞的 UI 做测试, 注意本项目根本

没做 UI，就是把获得信息扔到 mainview 上。具体布局代码请参照 activity_main.xml

可以被随便吐槽的界面如下：



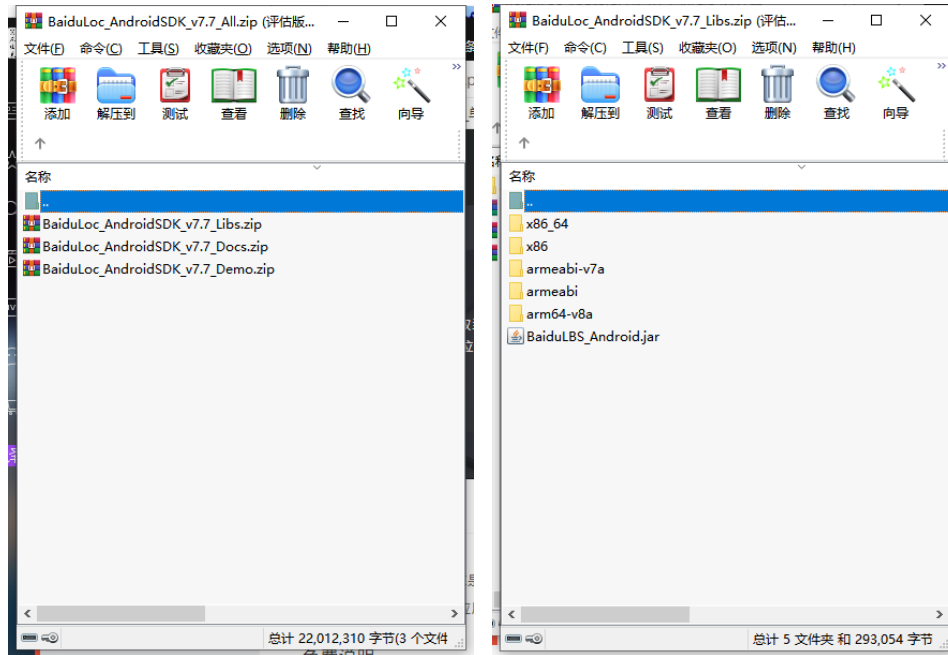
全是线性布局，另外 Refresh 功能没写，因为暂时用不到。

定位实现：基于百度 SDK，下载传送门如下：

<http://lbsyun.baidu.com/index.php?title=android-locsdk>

所用的是定位 SDK 这个功能很强大，高德也有类似的东西，更强大的功能可以参考开发文档，另外不建议阅读百度文档，写的太烂，看高德吧，文笔好还有动画。为了防止功能不够，我下的全量版，使用的时候打开 LIBS.ZIP，LIBS 目录如最后所示，一个 jar 其

他全是文件夹。



Jar 放进 app/libs 其他文件夹在 app/src/main 下建一个 JniLibs 丢进去。然后 app 的 build.gradle 中加入本地仓库代码如下：

```
buildTypes {
    release {
        minifyEnabled false
        proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
    }
}

//本地仓库
repositories {
    flatDir {
        dirs 'libs'
    }
}

dependencies {
    implementation fileTree(include: ['*.jar'], dir: 'libs')
    implementation 'com.android.support:appcompat-v7:28.0.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
```

然后去 manifest 里加入权限

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.morsak.baidusdk">
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <!-- 这个权限用于访问GP定位 -->
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <!-- 用于访问wifi网络信息，wifi信息会用于进行网络定位 -->
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
    <!-- 获取运营商信息，用于支持提供运营商信息相关的接口 -->
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <!-- 这个权限用于获取wifi的获取权限，wifi信息会用来进行网络定位 -->
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
    <!-- 用于读取手机当前的状态 -->
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <!-- 写入扩展存储，向扩展卡写入数据，用于写入离线定位数据 -->
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <!-- 访问网络，网络定位需要上网 -->
    <uses-permission android:name="android.permission.INTERNET" />
    <!-- SD卡读取权限，用户写入离线定位数据 -->
    <uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
    <application
        android:networkSecurityConfig="@xml/network_security_config"

```

Application 里加入与 sdk 有关的代码

```

</intent-filter>
</activity>

<meta-data
    android:name="com.baidu.lbsapi.API_KEY"
    android:value="i7SDZgkQWpFGU06pncv58GGC28G1hIi1" >
</meta-data>
<!-- 定位 -->
<service
    android:name="com.baidu.location.f"
    android:enabled="true"
    android:process=":remote" />

</application>
</manifest>

```

记得申请 key 把 value 代换。

在 res 里新建一个 network_security_config.xml 里面具体代码参看文件，修改网络安全协议，然后回到 manifest 里加入

```

<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
<application
    android:networkSecurityConfig="@xml/network_security_config"
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"

```

然后 com.morsak.baidusdk 包里有四个文件夹 data、entity、impl、ui

其中 data 中定义了用于 json 的格式代码，天气数据源采用的是

<https://www.juhe.cn/docs/api/id/73>

你可以用他的接口申请 key 用他规定的形式发送一下请求你就知道 data 里的 weatherdatabean 为什么那么定义了。

Entity 是常量类，是我做测试用的，另外记录了我申请的天气数据 key

Impl 是接口里面按照 retrofit 的方法写了天气获取方法，retrofit 详细请看 <http://square.github.io/retrofit/>

这个东西现在被谷歌接收了，属于谷歌安卓项目，写的很好。

Main 文件里，最后的数据在下图位置

```
// 请求天气数据
private void requestWeather(String city)
{
    Call<WeatherDataBean> weatherBeanCall = impl.getWeather(city, Constans.WEATHER_KEY);
    weatherBeanCall.enqueue(new Callback<WeatherDataBean>() {
        @Override
        public void onResponse(Call<WeatherDataBean> call, Response<WeatherDataBean> response) {
            //Log.i(Constans.TAG, response.body().toString());
            WeatherDataBean bean = response.body();
            Calendar cal;
            cal = Calendar.getInstance();
            cal.setTimeZone(TimeZone.getTimeZone("GMT+8:00"));
            String hour, minute, second;
            if (cal.get(Calendar.AM_PM) == 0)
                hour = String.valueOf(cal.get(Calendar.HOUR));
            else
                hour = String.valueOf(cal.get(Calendar.HOUR)+12);
            minute = String.valueOf(cal.get(Calendar.MINUTE));
            second = String.valueOf(cal.get(Calendar.SECOND));
            mTvCity.setText(bean.getResult().getCity());
            mTvUpdateTime.setText(hour+":"+minute+":"+second);
            mTvCode.setText(bean.getResult().getRealtime().getTemperature());
            mTvWeather.setText(bean.getResult().getRealtime().getInfo());
            mTvWind.setText("风向|"+bean.getResult().getRealtime().getDirect());
            mTvWindCode.setText("风力|"+bean.getResult().getRealtime().getPower());
        }
    });
}
```

@Override

获取是异步的，所以基本应该不担心效率，我把更新时间也写这里了，你可以拿出去单个做一个函数，bean 对应的就是使用 weatherdatabean 格式但是带着返回数据的 weatherdatabean 实例，它的数据获取方式如上图，你只要根据结构使用相应的 get 就好。

最后顺带一提的是，如果你的项目不使用了，请关闭百度和聚合数据的项目申请 key 来提升安全性。

还有如果请求返回的结果是错误的，或者根本没有返回数据，请检查天气接口的返回数据是否有误，可能天气那边的请求到达上限了，你会得到一个 error 值，请求上限是 500 次，做展示应该够了，你可以注册会员或者尝试新的接口，只要替换相应的 json 格式就好，即 weatherdatabean 的格式规定。