

手机卫士第一天笔记

代码的组织结构

推荐使用根据功能模块划分

- Activity com.itheima.mobilesafe.activity
- 后台服务 com.itheima.mobilesafe.service
- 广播接受者 com.itheima.mobilesafe.receiver
- 数据库 com.itheima.mobilesafe.db.dao
- 对象(java bean) com.itheima.mobilesafe.domain/bean
- 自定义控件 com.itheima.mobilesafe.view
- 工具类 com.itheima.mobilesafe.utils
- 业务逻辑 com.itheima.mobilesafe.engine

在创建项目时的选项解释

- minimum SDK 要求最低的安装版本, 安装apk前,系统会判断当前版本是否高于(包含)此版本, 是的话才允许安装
- maxSdkVersion 要求最高的安装版本(一般不用)
- Target SDK 目标SDK, 一般设置为开发时使用的手机版本, 这样的话,系统在运行我的apk时,就认为我已经在该做了充分的测试, 系统就不会做过多的兼容性判断, 从而提高运行效率
- Compile With 编译程序时使用的版本

代码获取软件的versioncode和versionname

```
PackageManager packageManager = getPackageManager();//获取包管理器
```

```
try {
```

```
    PackageInfo packageInfo = packageManager.getPackageInfo(getPackageName(),  
PackageManager.GET_CONFIGURATIONS);//获取包的信息
```

```
    int versionCode = packageInfo.versionCode; //获取版本号
```

```
    String versionName = packageInfo.versionName; //获取版本的名称号
```

```
    splash_tv_version.setText("版本:" + versionName);//将获取到的版本名称号显示到splash界面上
```

```
} catch (NameNotFoundException e) {
```

```
    e.printStackTrace();
```

```
}
```

设置文本的阴影效果

```
<TextView
```

```
    android:id="@+id/splash_tv_version"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:layout_centerHorizontal="true"
```

```
    android:layout_centerVertical="true"
```

```
    android:textSize="20sp"
```

```
    android:shadowColor="#ff0000"
```

```
    android:shadowDx="1"
```

```
    android:shadowDy="1"
```

```
android:shadowRadius="10"
```

```
android:text="@string/test_text"
```

```
android:textColor="#ffffff" />
```

在模拟器或者真机的调试中, 如果要访问上位机的资源, 那么上位机的ip地址为: 10.0.0.2

从网络上获取数据的一般步骤

```
URL url; //创建url对象
```

```
try {
```

```
    url = new URL("http://10.0.2.2:8080/update.json");
```

```
    HttpURLConnection connection = (HttpURLConnection)url.openConnection(); //获取连接对象
```

```
    connection.setConnectTimeout(5000); //设置连接超时
```

```
    connection.setReadTimeout(5000); //设置请求超时
```

```
    connection.setRequestMethod("GET"); //设置请求方法
```

```
    connection.connect(); //开始连接
```

```
    if (connection.getResponseCode() == HttpURLConnection.HTTP_OK)
```

```
    {
```

```
        InputStream in = connection.getInputStream();
```

```
        //处理获取到的数据
```

```
    }
```

```
} catch (MalformedURLException e) { //url异常, 即网络异常
```

```
    e.printStackTrace();
```

```
} catch (IOException e) { //io异常
```

```
    e.printStackTrace();
```

```
}
```

涉及网络的耗时操作必须放到了子线程中去执行, 否则会导致主线程阻塞, 如果主线程阻塞时间超过5秒, 所在的activity就会崩溃
