客户端静态代码扫描流程及工具说明

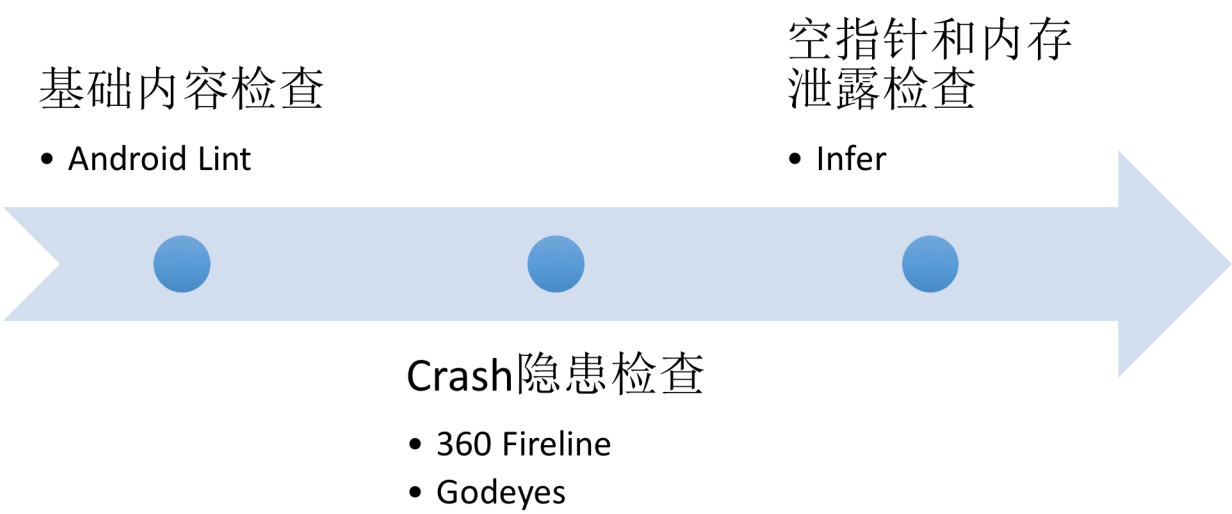
0. 更新说明

撰写人	版本	描述
马季	0.1	初稿

1. 扫描流程

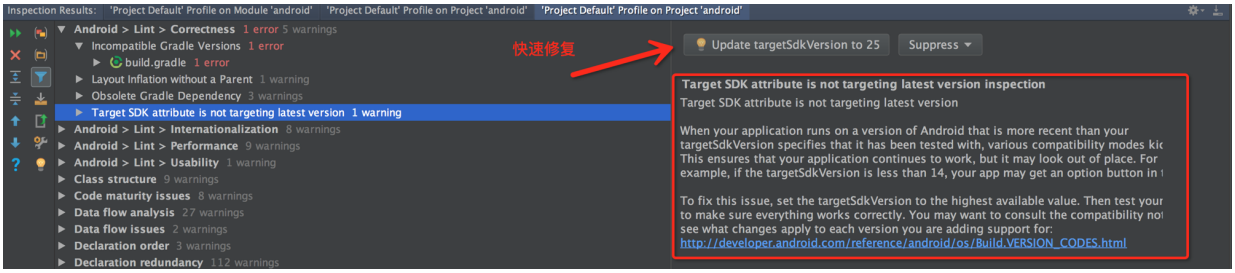
1.1 Android 客户端扫描流程

分为3个阶段：



1.1.1 基础内容扫描

使用 **Android Lint** 对项目进行扫描，该工具已将扫描到的问题进行了分组，同时定义了问题的严重级别： `error` 和 `warning` 。在 **Android Studio** 的 **Inspection Results** 视图窗中，点击问题标题即可在右边的详情视图中查看该问题的具体解释等内容，针对部分内容还有直接进行自动修复的按钮，如图：



由于检查的内容繁多，我们重点关注以下几个**问题组**的相关内容:

- 以 **Android** 开头的组，例如
    - **Android > Lint > Correctness** (可能影响程序正确性)
    - **Android > Lint > Performance** (可能影响程序性能)
    - **Android > Lint > Security** (可能影响程序安全性)
    - 等等
- **Class structure** 组：指出类的设计上可能存在的问题
- **Code style issues** 组：有助于提供代码书写规范
- **Probable bugs** 组：有助于发现隐藏的问题

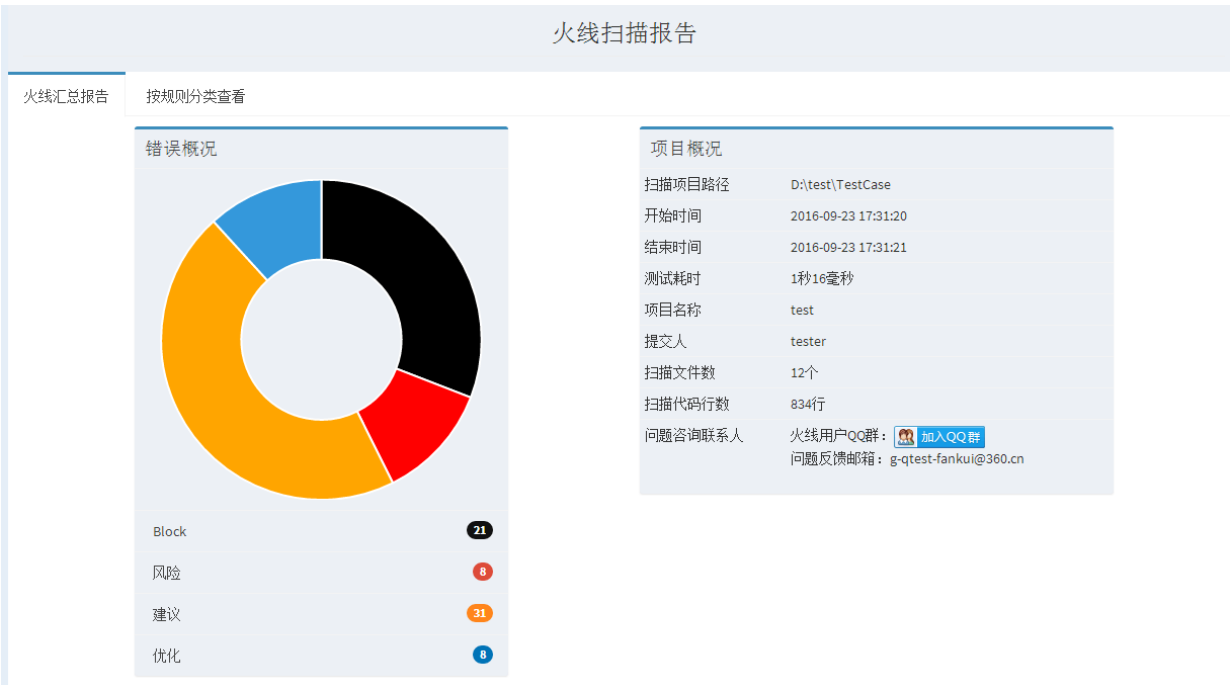
> 检查通过标准:
>
> 上述的列出的问题组别**不出现**或者**出现但只包含** `warning` **类型的问题**

1.1.2 可能引起Crash的问题扫描

使用**360火线**和**Godeyes**对项目进行扫描，下面将分别说明二者扫描时的关注点和检查通过标准:

360火线

360火线共有61个检查项，按级别分为 `Block` 、 `风险` 、 `建议` 和 `优化` ,检查报告以html文件输出，在**按规则分类查看**的Tab中，可以查看具体问题位置及示例代码。如图



Show 25 entries Search:

风险类型	规则名称	等级	错误数量	规则描述	查看详情
数据泄露	应用程序备份恢复隐患	风险	1	建议将AndroidMannifest.xml文件android:allowBackup属性设置为false。当allowBackup标志值为true时，攻击者可通过adb backup和adb restore来备份和恢复应用程序数据。	

所在位置：D:\branch0224\PluginBase\build\intermediates\manifests\aapt\release\AndroidManifest.xml
元素名：application
错误元素位置：15

```
8                android:minSdkVersion="14"
9                android:targetSdkVersion="19" />
10
11           <application
12               android:allowBackup="true"
13               android:icon="@drawable/ic_launcher"
14               android:label="@string/app_name"
15               android:theme="@style/AppTheme" >
16           </application>
17
18       </manifest>
```

空指针	存在空指针引用	风险	1	此位置存在空指针引用,会导致空指针异常。	
空指针异常	破坏空判断	风险	1	如果自身抛出空指针异常空检查就会遭到破坏，比如你使用 ||代替 &&，反之亦然，如（应是&&）：if (string!=null || !string.equals("")){}。	

> 检查通过标准:
>
> 扫描结果中不出现`Block`、`风险`两类问题

Godeyes

Godeyes共检查23个错误，扫描结果以html文档的方式输出，文档中包含了检查问题的**描述**、**示例**以及**推荐方案**，方便理解。值得注意的是在扫描结果的显示上，报告**只会给出问题所在的行号**。如图

GodEyes 扫描结果

扫描规则	扫描结果
dismiss()方法调用前isShowing未判断的隐患	22
数组下标越界隐患	19
格式化数字异常未捕获的隐患	6
使用String.substring前未判断String长度隐患	4
使用String.split结果未判断长度隐患	2
使用除法或求余没有判断分母长度隐患	2
系统API兼容性隐患	1
使用IO流后没有关闭导致OOM隐患	1
ArrayList使用get方法获取元素未判断下标有效性的隐患	1
复写生命周期函数没有调用super函数隐患	0
主动抛出异常未捕获处理隐患	0
大图片解析导致OOM的隐患	0
通过HashMap获取对象使用未判空隐患	0
Activity未在AndroidManifest.xml中注册隐患	0
销毁Dialog前是否isShowing未判断隐患	0
查询数据库没有关闭游标导致OOM的隐患	0
ArrayList对象使用未判空隐患	0
使用Bundle与使用从Bundle获取到的数据未判空隐患	0
方法中存在return null返回对象直接进行方法调用隐患	0
使用动态载入界面的元素未判断是否属于此界面的隐患	0
使用在intent中获取的数据前未判空隐患	0
添加Fragment前未判断是否IsAdded隐患	0
数据库操作异常未捕获处理隐患	0

1 Android

1.1 【NullPointerException】dismiss()方法调用前isShowing未判断的隐患

1.1.1 摘要

问题：在调用系统的dismiss()方法前，没有对状态进行判断，导致抛出NullPointerException异常。
解决方案：在调用系统的dismiss()方法时，需要对状态先进行判断。

1.1.2 示例

在下面代码中，popupWindow.dismiss()调用时出现了NullPointerException。

```
if (mContext != null &&!((Activity)mContext).isFinishing())
{
  popupWindow.dismiss();
  setFocusable(false);
 }
```

1.1.3 推荐方案

在调用系统的dismiss()方法前先进行isShowing的判断，修改如下：

```
if (mContext != null &&!((Activity)mContext).isFinishing() && isShowing())
{
  popupWindow.dismiss();
  setFocusable(false);
 }
```

1.1.4 扫描结果

Java文件名	行号
com/unionpay/mobile/android/widgets/UPWidgetKeyBoard.java	149
com/unionpay/mobile/android/widgets/UPPromotionWidgetHalf.java	46,55,70,86,97
com/unionpay/mobile/android/widgets/UPAreaCodeWidget.java	66,78
com/unionpay/mobile/android/widgets/UPDropDownWidget.java	53,66
com/unionpay/mobile/android/ui/UPStyleWindow.java	132
com/unionpay/mobile/android/widgets/UPWidgetKeyboardSimple.java	157
com/unionpay/mobile/android/widgets/UPPromotionItem.java	68,77,92,103,119
com/unionpay/mobile/android/widgets/UPPromotionWidget.java	46,61,77
com/unionpay/mobile/android/widgets/UPCertTypeWidget.java	62,74

> 检查通过标准:
>
> 各扫描项扫描结果为0

1.1.3 空指针和资源泄露扫描

使用Infer工具对**可能的空指针**和**可能的资源泄露**进行扫描，Infer工具会在项目的根文件夹下生成`infer-out`的文件夹，重点关注`bugs.txt`这个文件，文件中会详细指出可能存在的问题的代码片段及相应的解释，示例如下：

```
Found 70 issues

PluginBase/src/com/unionpay/mobile/android/resource/ResourceManager.java:219: er
   resource of type `java.io.DataInputStream` acquired to `dis` by call to `new(
**Note**: potential exception at line 164
  217.                     dis.close();
  218.                     is.close();
  219. >                } catch (IOException e) {
  220.                     e.printStackTrace();
  221.                     dr = null;
```

```
PluginBase/src/com/unionpay/mobile/android/upviews/UPRuleView.java:402: error: N
  object returned by `getItemByName("instalment")` could be null and is derefere
  400.            } else {
  401.                ((UPDropDownWidget) getItemByName(Rules.TYPE_INSTALMENT))
  402. >                      .setmCanShow(true);
  403.                ((UPDropDownWidget) getItemByName(Rules.TYPE_INSTALMENT))
  404.                      .onCheckBoxStatusChanged(true);
```

> 检查通过标准:
>
> 对检查的问题尽量修复或者编写保护语句避免抛出异常。

2. 工具使用

2.1 Android 客户端

以下内容均以 **Android Studio** 为默认的开发环境

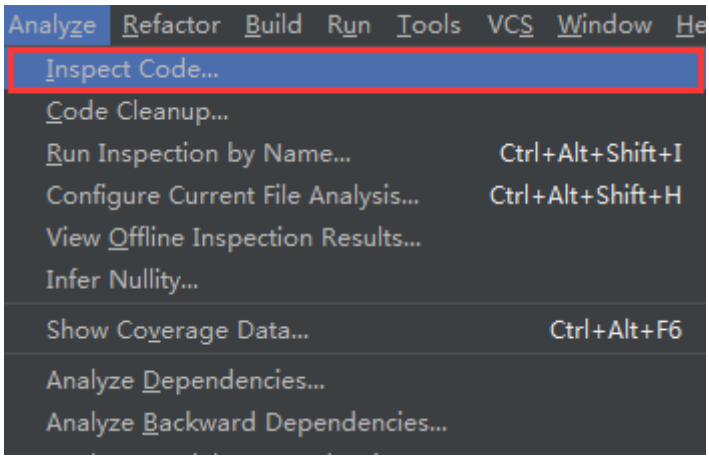
2.1.1 Android Lint

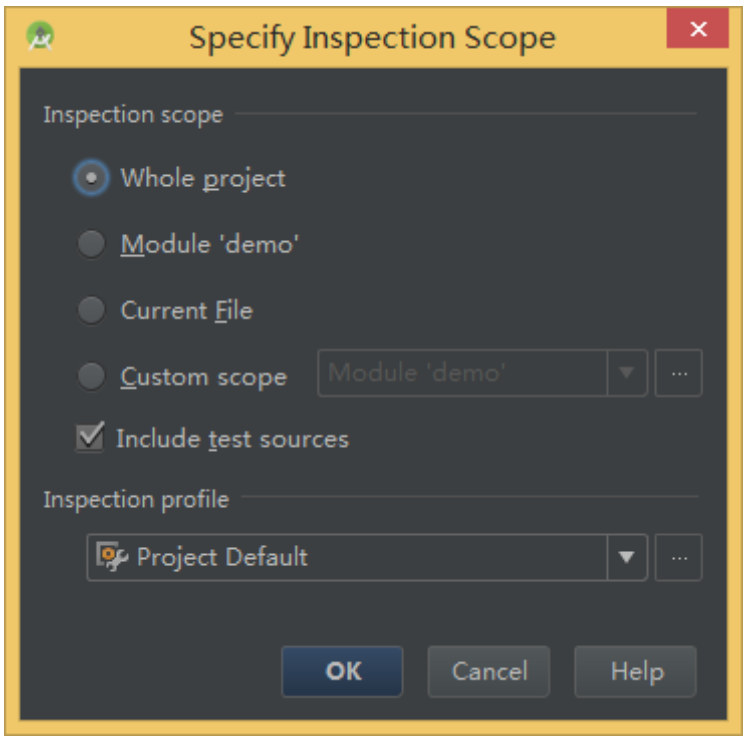
该工具已经默认集成 **Android Studio** 中

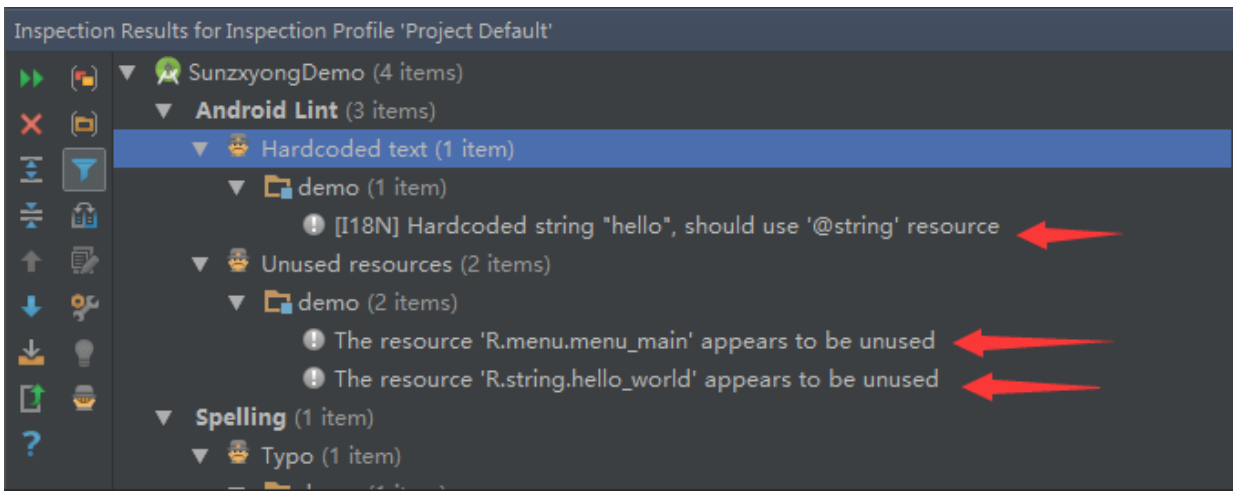
使用方法:

Android Stuido -> 菜单栏 -> **Analyze** -> **Inspect Code** -> 根据需要选择相应的扫描范围 -> **OK** -> 启动扫描

如图:







参考资料

- http://tools.android.com/tips/lint/(需要翻墙)
- http://www.bubuko.com/infodetail-1055648.html
- android-studio-中使用-lint

- Android APK瘦身之Android Studio Lint (代码审查)

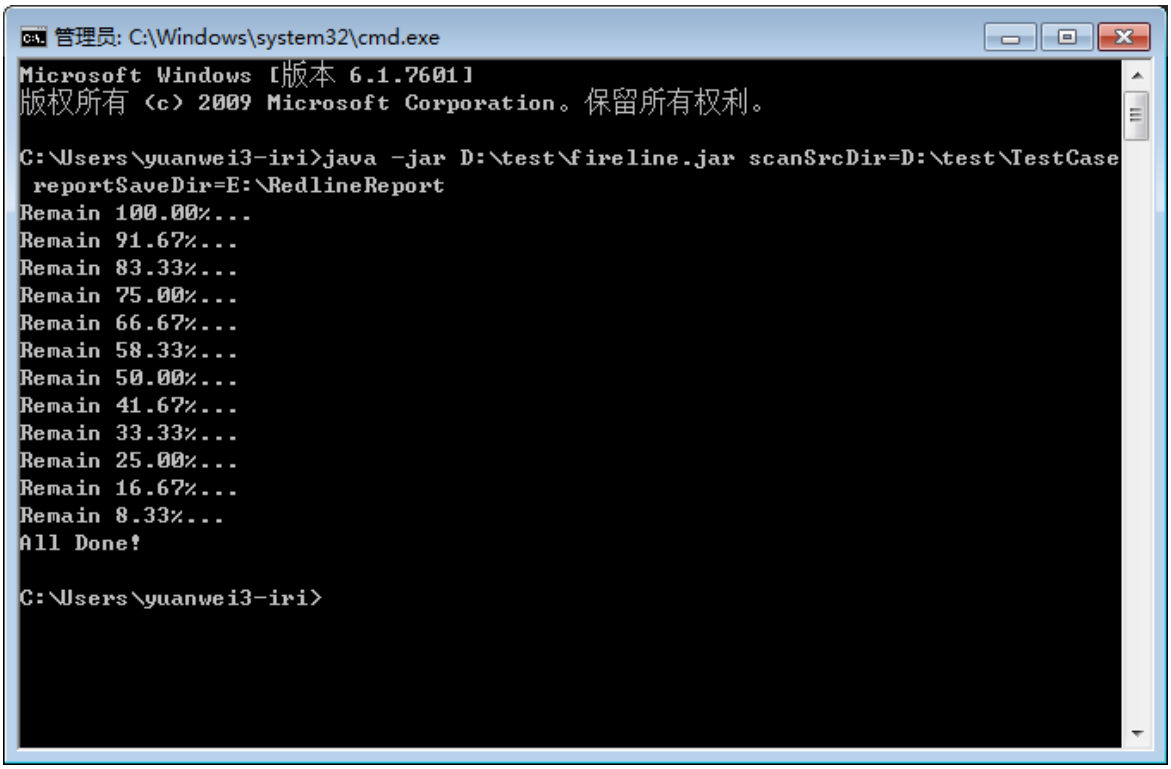
2.1.2 360 火线

使用方法

> 详情参考 官方使用方法 火线插件目前可以在**Android Studio**中进行在线搜索安装。

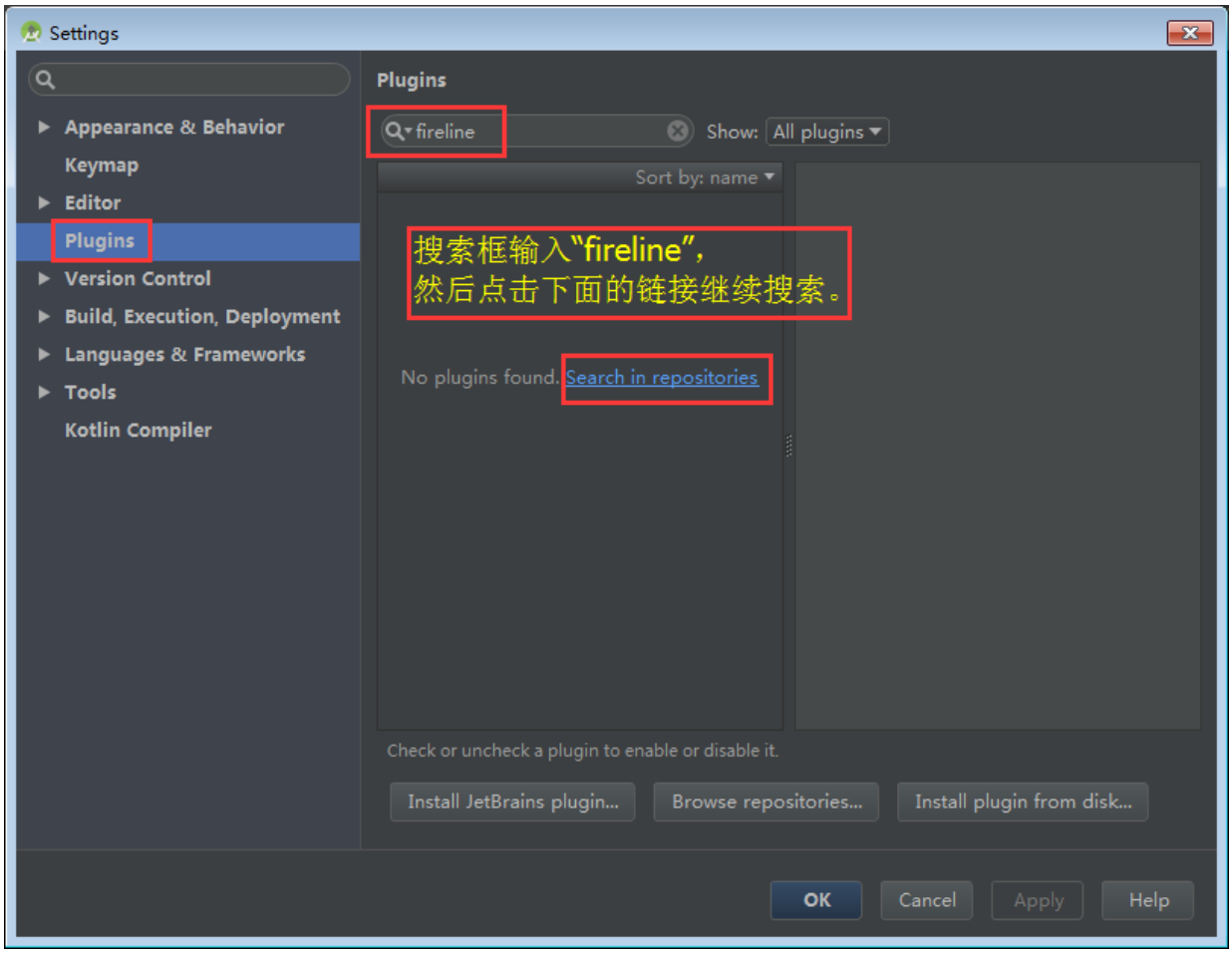
- jar包版本使用

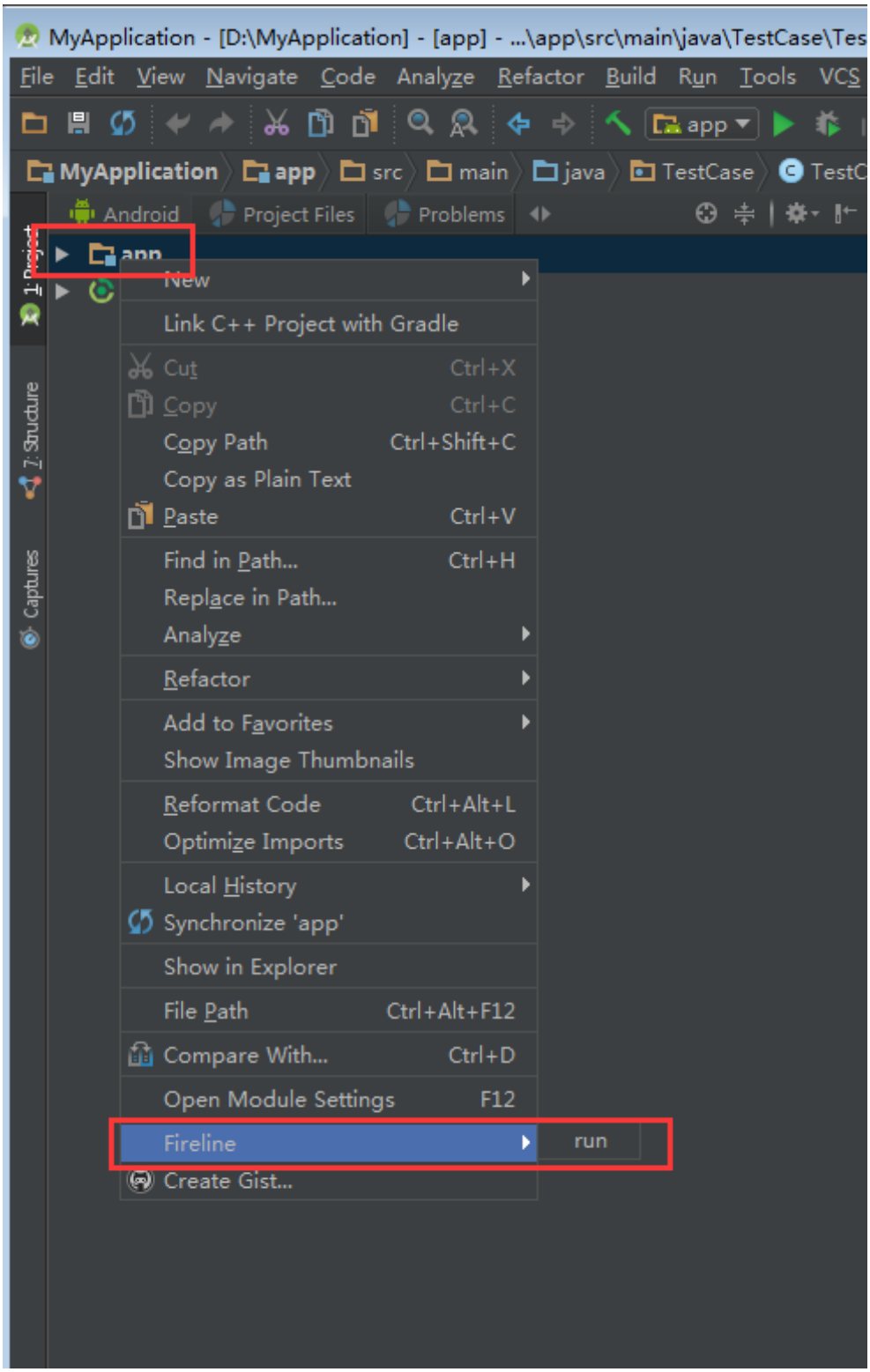
```
java -jar D:\test\fireline.jar -s=D:\test\TestCase -r=E:\RedlineReport
// 参数解释:
//【必填项】-s或scanSrcDir为被扫描的项目工程路径
//【必填项】-r或reportSaveDir为火线报告输出路径
```



- Android stuido版本

  i. **Android Studio** -> 菜单栏 -> **File** -> **Settings...** -> **Plugins**
  ii. 搜索框 -> 搜索**fireline** -> **install** -> 重启
  iii. 使用 -> Project视图 -> 鼠标右键 -> **fireline** -> **run** 生成报告


Settings
Appearance & Behavior
Keymap
Editor
Plugins
Version Control
Build, Execution, Deployment
Languages & Frameworks
Tools
Kotlin Compiler
Plugins
fireline
Show: All plugins
Sort by: name
搜索框输入“fireline”，
然后点击下面的链接继续搜索。
No plugins found. Search in repositories
Check or uncheck a plugin to enable or disable it.
Install JetBrains plugin...
Browse repositories...
Install plugin from disk...
OK
Cancel
Apply
Help



参考资料

- http://magic.360.cn/

2.1.3 Godeyes

使用方法

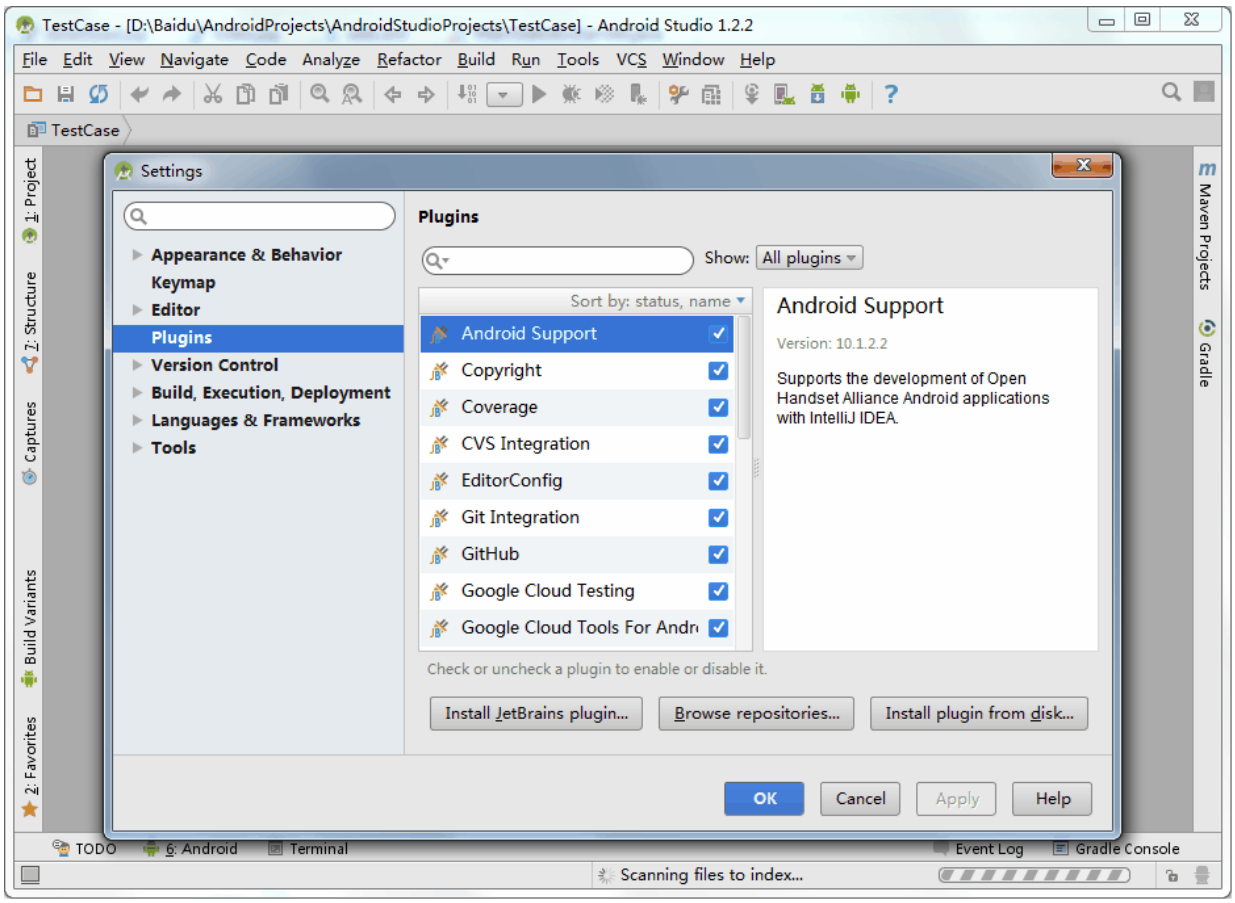
> 详情参考 官方使用方法

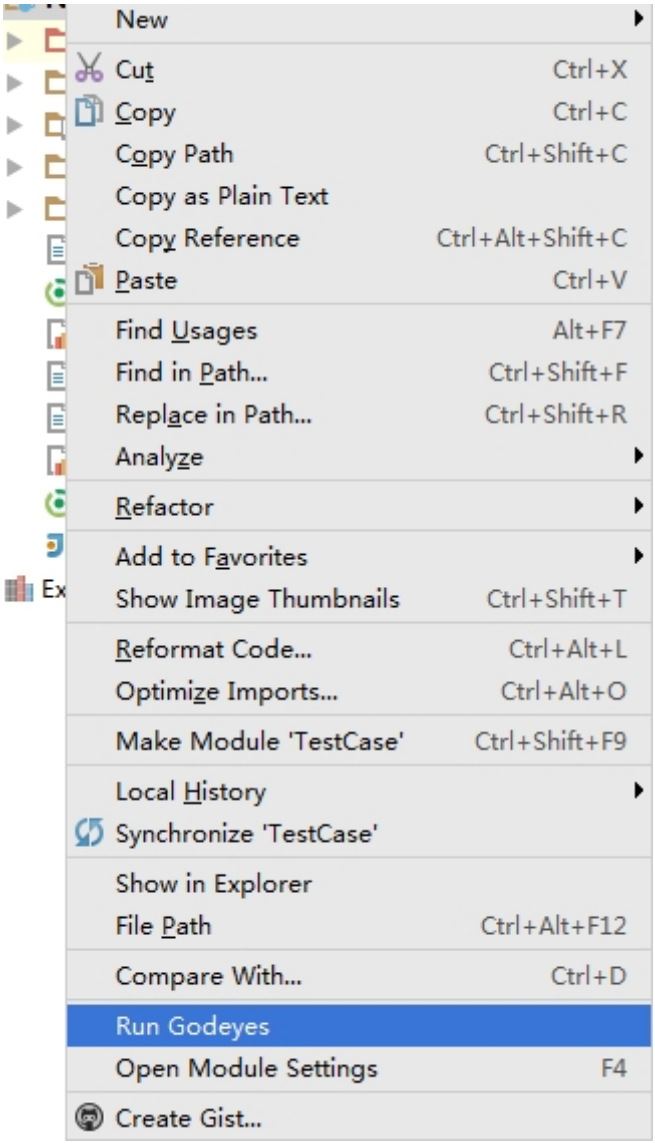
1. 下载 Android Studio版本插件 下载地址

2. `Android Studio` -> 菜单栏 -> `File` -> `Settings` -> `Plugins`

3. 选择`Install plugin from disk` -> 选择已下载的`Godeyes_Android_Vx.x_(for_AndroidStudio).zip` -> `OK` -> 安装完成 -> 重启

4. Project视图 -> 鼠标右键 -> `Run Godeyes` -> 生成报告





参考资料

- http://godeyes.duapp.com/
- http://blog.csdn.net/xwh_1230/article/details/51312847

2.1.4 Infer

> 注意:
>
> 1. 仅支持 Mac 和 Linux 环境
> 2. 需要Python 且 Python >= 2.7

使用方法

- 安装，请参考 https://infer.liaohuqiu.net/docs/getting-started.html
- 使用方法

```
cd {项目的根目录}
./gradlew clean
infer -- ./gradlew build
```

一段时间后会在项目的根目录下生成infer-out这个文件夹，里面的bugs.txt文档里记录的就是扫描出的问题。

参考资料

- https://infer.liaohuqiu.net/
- http://blog.csdn.net/itfootball/article/details/46474235
- https://github.com/facebook/infer/blob/master/INSTALL.md
- https://github.com/facebook/infer/releases