

**软件工程(I)**

**小米便签开源代码**

**质量分析报告**



**组长学号姓名： 210340170 薛皓天**

**成员学号姓名： 210340169 薛傲翔**

**210340139 陈政杨**

**210340157 刘薇**

## 二〇二三年五月

小米便签开源软件是一款较为流行的备忘录软件，其由Java语言编写，基于Android操作系统进行开发运行和维护。小米便签开源软件规模适中，通过初步阅读分析其代码质量较高，故对其进行进一步的质量分析和审查。

### 代码质量分析方法

对开源软件代码的质量进行分析旨在理解高质量的软件系统在代码层面和设计层面应具有哪些方面的特征，掌握编写高质量程序代码的方法和技巧，同时发现开源软件代码中尚存的质量问题。对开源软件代码的质量分析可以采用以下二种方法：人工分析和自动分析。

1. **人工**

静态测试时指测试程序不在机器上运行，而是采用人工检测和计算机辅助静态分析的手段对程序进行检测。静态分析中进行人工测试的主要方法有桌前检查（Desk Checking）、代码审查和代码走查。经验表明，使用这种方法能够有效发现30%-70%的逻辑错误。

值得说明的是，使用静态测试的方法也可以实现白盒测试。例如：使用人工检查代码的方法来检查代码的逻辑问题，也属于白盒测试范畴。

1. **自动**

（1）CodeClimate：是一个基于云的代码质量平台，它提供了用于检测和管理代码质量的工具和服务。与SonarQube相似，CodeClimate对各种编程语言提供丰富的代码规则和静态代码分析，支持多种CI/CD工具和版本控制系统。

（2）ESLint：是一个流行的JavaScript代码检查工具，它支持自定义规则，可以用于检查代码错误、风格问题和可维护性问题等。ESLint可以与许多编辑器和IDE集成，提供实时代码检查和反馈。

（3）PMD：是一个Java代码分析工具，可以识别和管理代码错误、漏洞、缺陷等问题。它提供了多种分析规则和插件，可以用于检查代码质量，同时提供了多种可视化报告。

（4）Snyk：是一款针对安全漏洞和开源组件的代码检查工具，可以帮助团队及时发现和修复各种安全漏洞。它支持多种编程语言和集成开发环境，提供实时监控和报告。

（5）SonarQube：SonarQube是一款开源的代码质量管理工具，主要用于检测和管理代码的质量问题。它提供了一套全面的分析和管理工具，可以帮助团队持续提高代码质量，减少技术债务，同时提高开发者的生产力。SonarQube能够识别和管理代码错误、漏洞、缺陷、安全问题等各种质量问题，并提供实时分析和反馈，可以快速地为团队提供代码质量报告，帮助开发团队及时发现和解决问题。

本报告主要使用SonarQube工具进行针对小米便签开源软件进行质量分析，辅以少量的人工分析。

### SonarQube「代码质量分析工具」简介

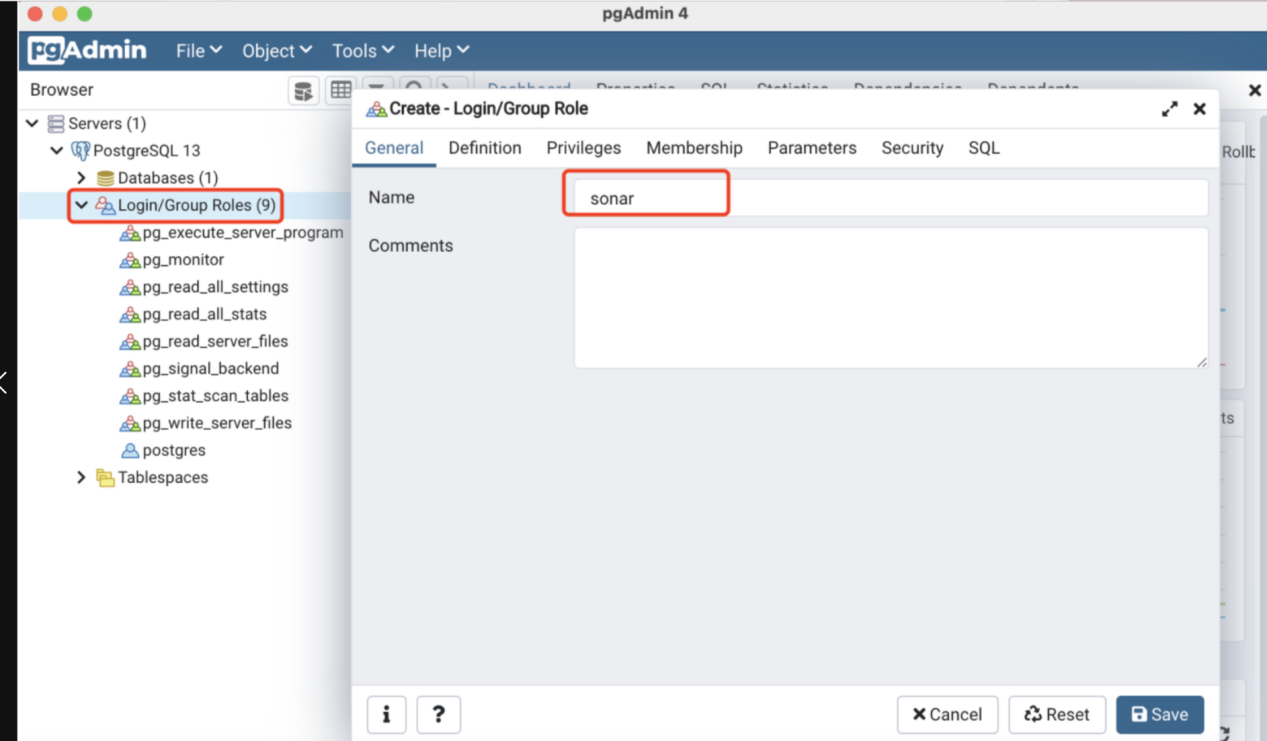
SonarQube是一个自我管理的自动代码审查工具，它系统地帮助您交付干净的代码。作为Sonar解决方案的核心元素，SonarQube集成到您现有的工作流程中，并检测代码中的问题，以帮助您对项目执行持续的代码检查。该工具分析了30多种不同的编程语言，并集成到您的CI管道和DevOps平台中，以确保您的代码符合高质量标准。编写干净的代码对于维护健康的代码库至关重要。我们将干净代码定义为满足特定定义标准的代码，即除了具有其他关键属性外，可靠、安全、可维护、可读和模块化的代码。这适用于所有代码:源代码、测试代码、作为代码的基础结构、粘合代码、脚本等。Sonar的Clean as You Code方法消除了许多因在开发过程的后期审查代码而产生的陷阱。当你的新代码(已添加或更改的代码)中有需要修复或审查的地方时，Clean as You Code方法会使用你的质量门来提醒/通知你，允许你保持高标准并专注于代码质量。

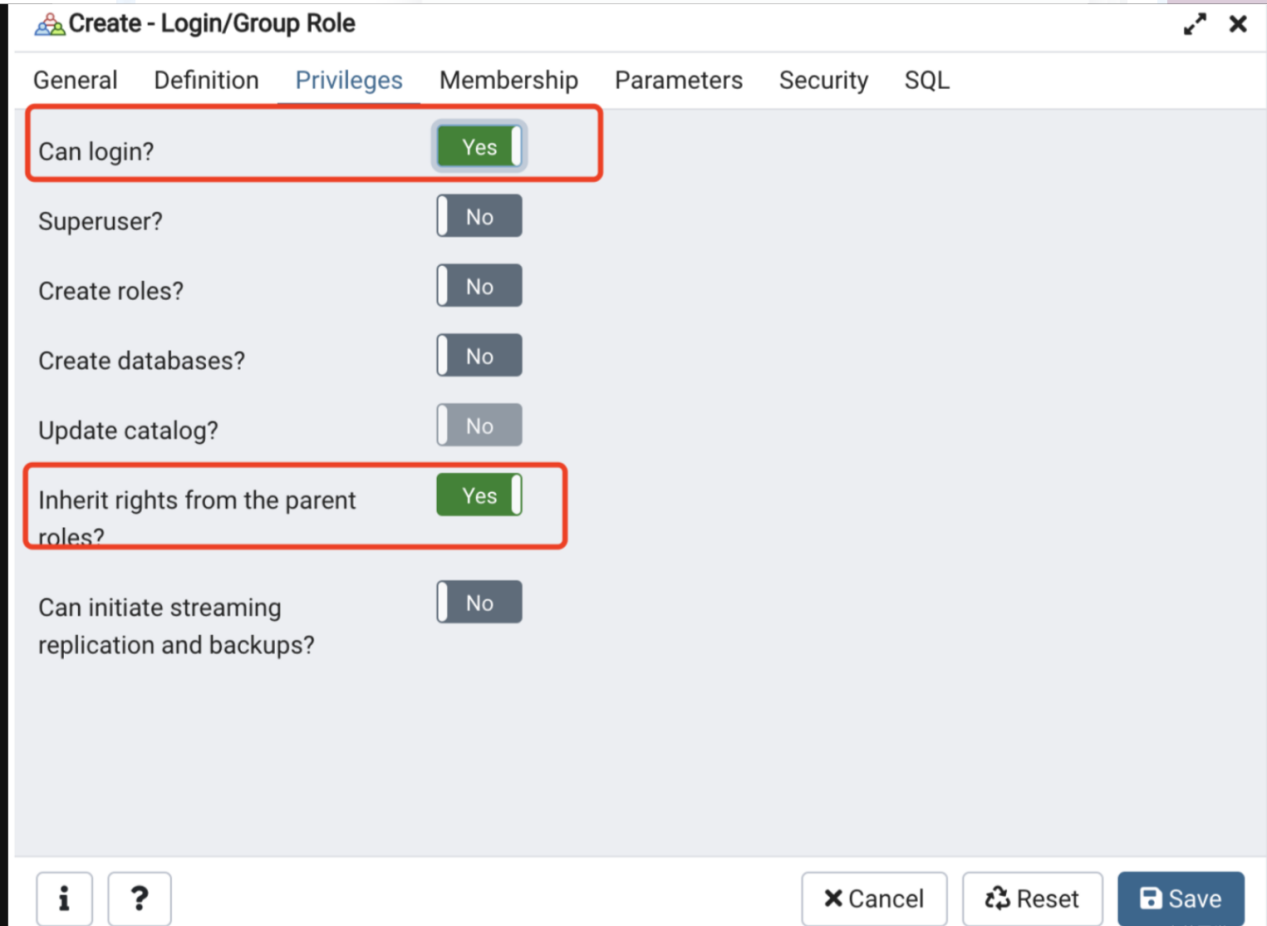
### SonarQube使用配置说明

1. 配置数据库

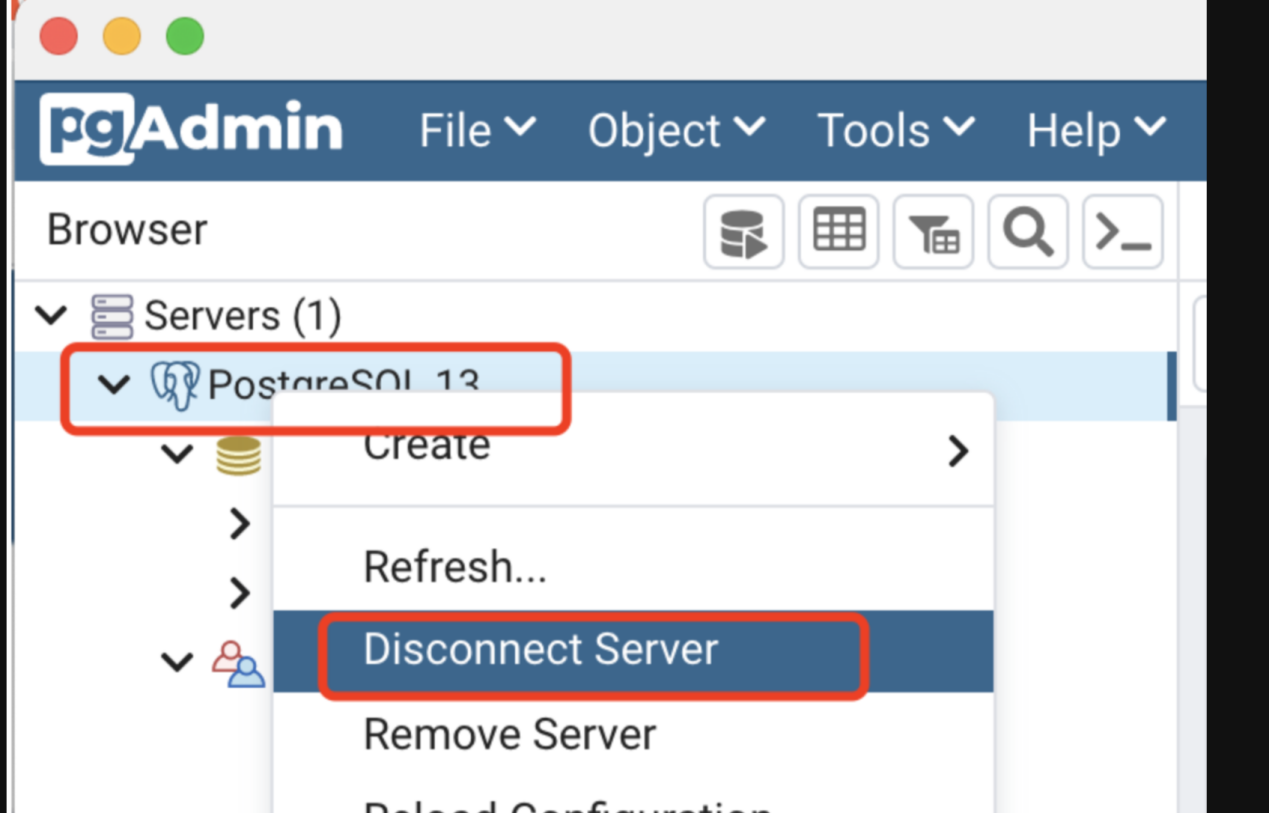
1.1创建sonar用户和sonar数据库

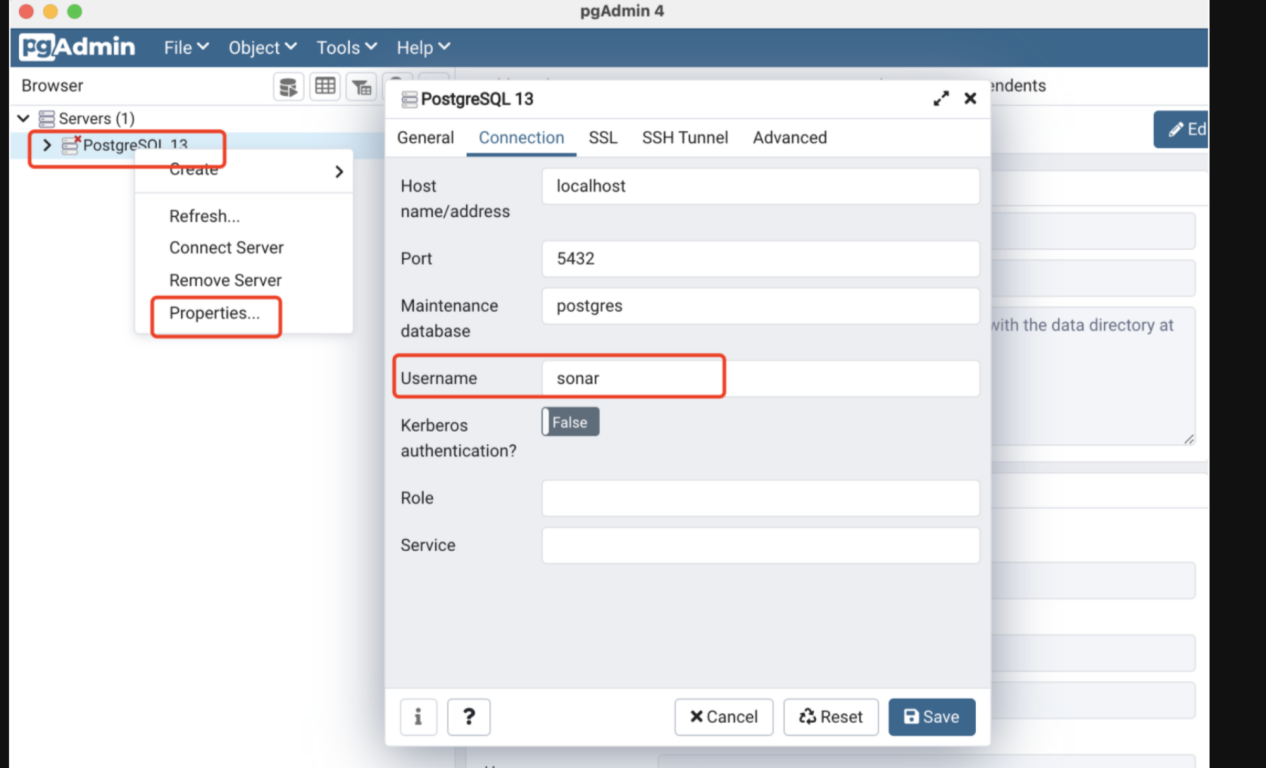
<1> 创建用户--sonar：双击Login/Group Roles——>在General中输入Name：sonar——>在Definition中输入密码：123456——>在Privileges中勾选

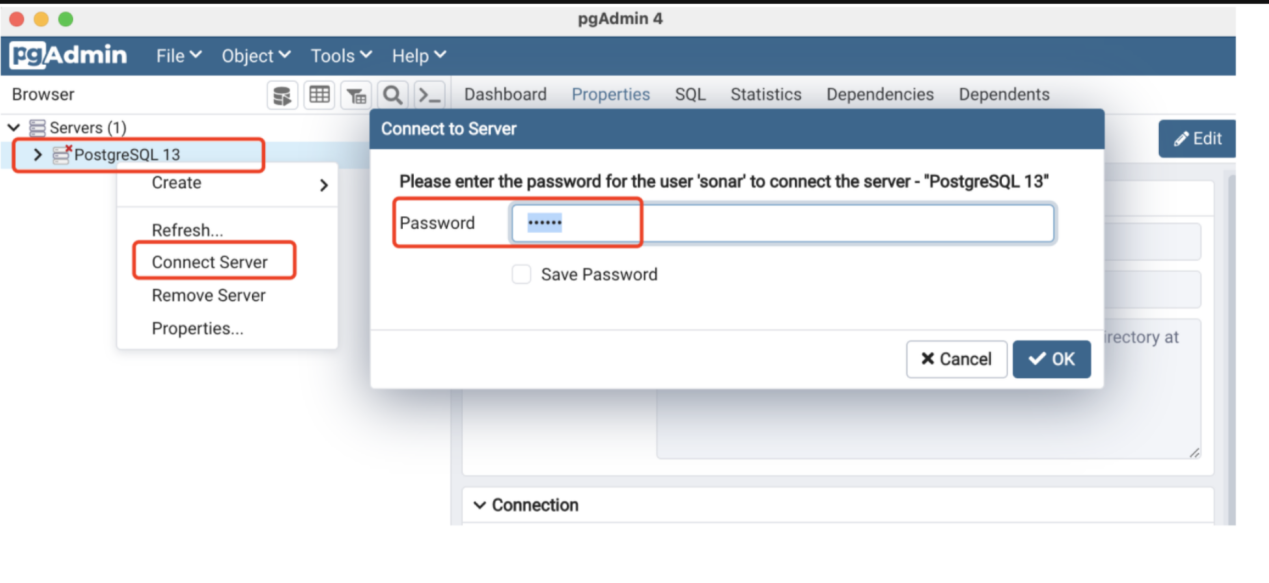




<2> 切换用户连接：右击PostgreSQL13 ——Disconnect Server







1. sonar的相关配置

2.1在sonar.properties文件中，需要输入如下信息：

sonar.jdbc.url=jdbc:postgresql://localhost/sonar?currentSchema=public # sonar表示数据库名字

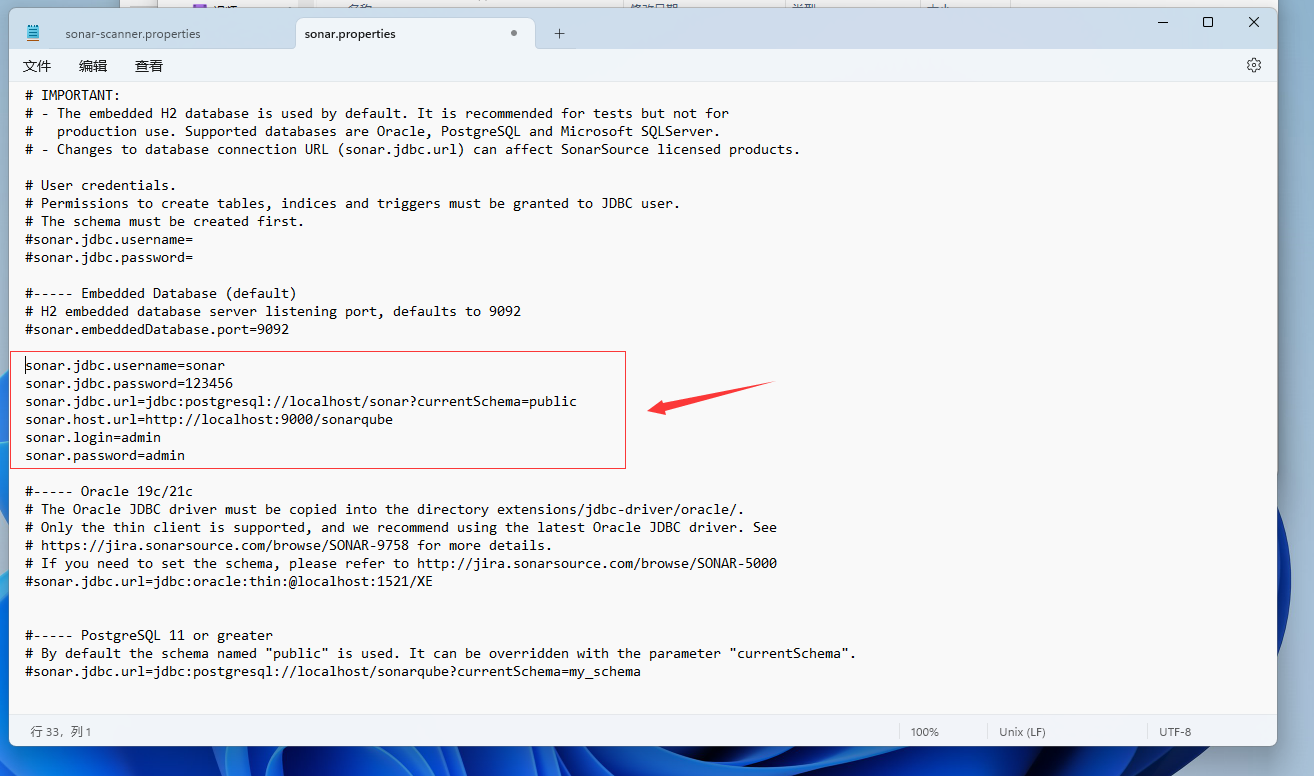
sonar.host.url=http://localhost:9000/sonarqube

sonar.jdbc.username=sonar #数据库用户名

sonar.jdbc.password=123456 # 密码

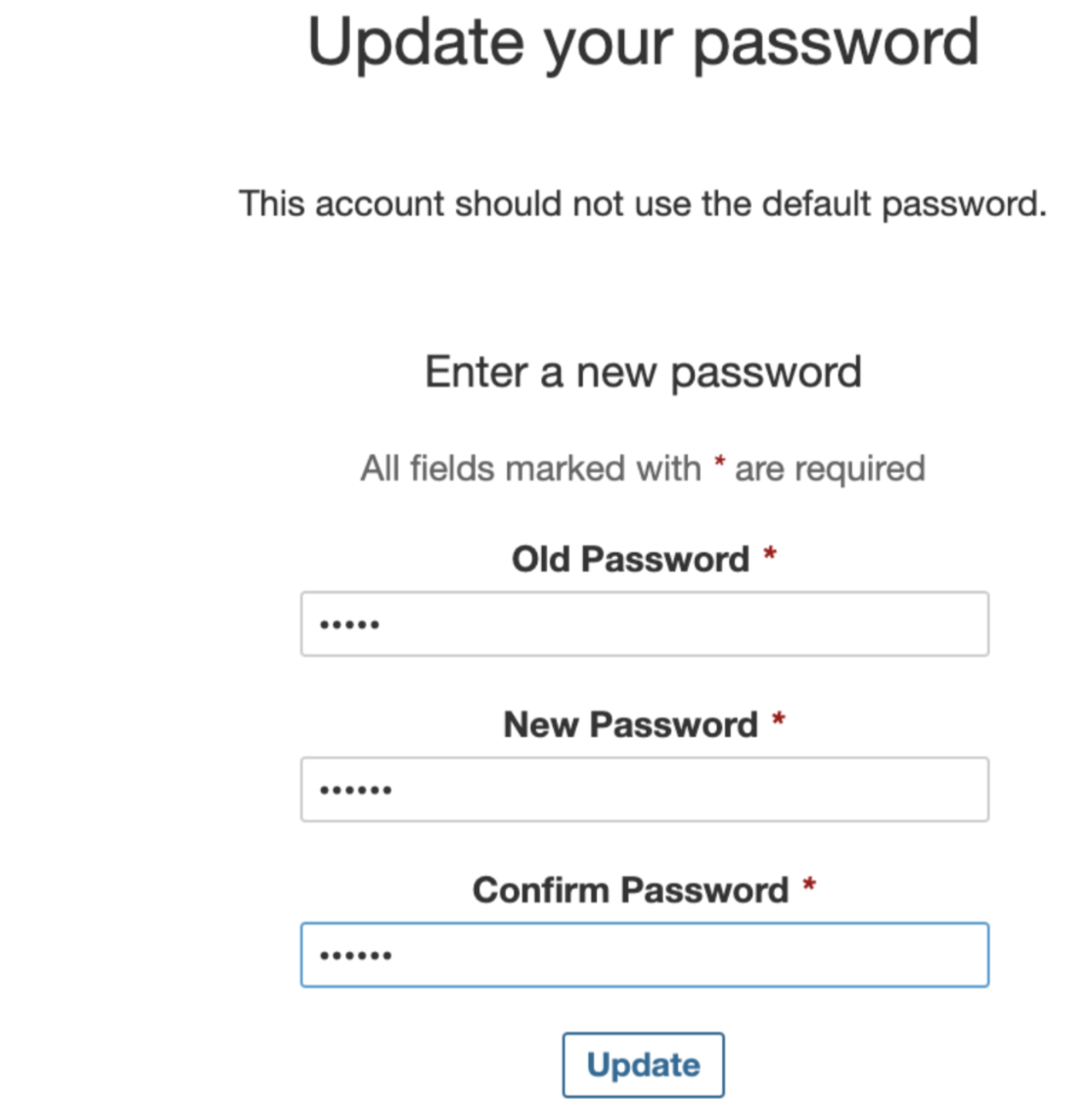
sonar.login=admin #登陆sonarqube的名字

sonar.password=admin #登陆sonarqube的密码

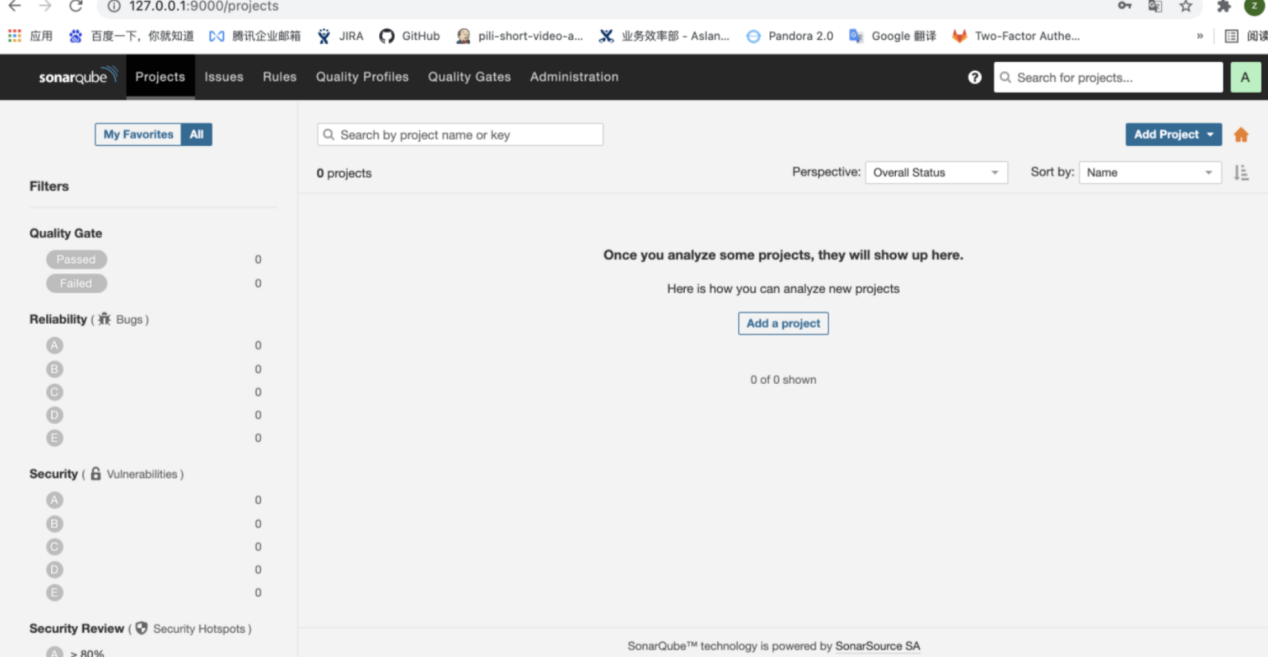


1. 启动sonarqube

找到StartSonar.bat并执行，然后再浏览器输入：http://127.0.0.1:9000,就会出现如下界面提示你更新一下密码：



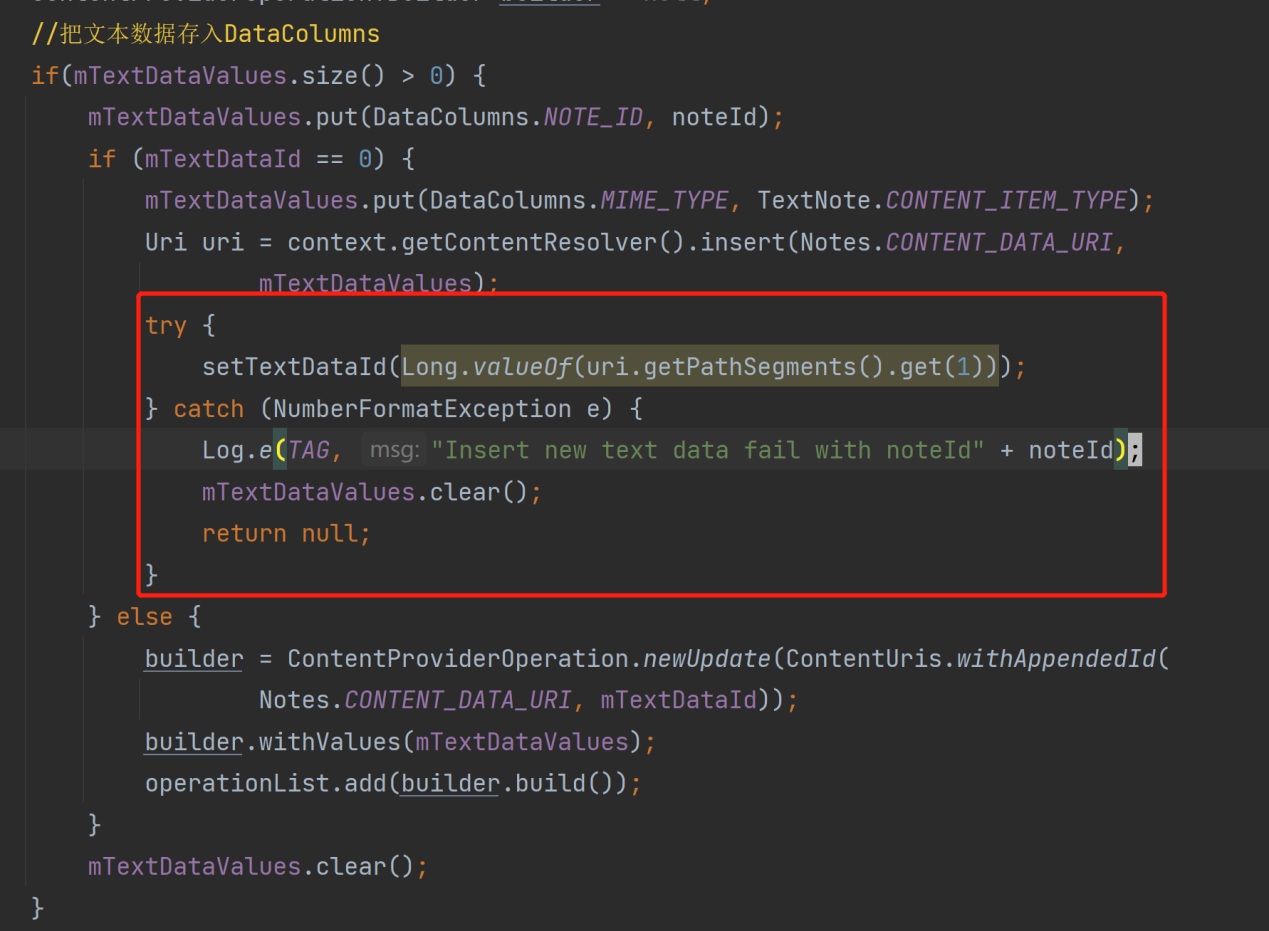
更新完毕后，看到如下界面，代表着启动成功



### 通过人工方式分析小米便签的质量情况

通过对“小米便签”开源软件代码的阅读和分析，可以发现“小米便签”程序代码蕴含有诸多值得学习的编程方法和技巧、编写高质量代码的规范和经验等。如“\*\*\*”类中方法“\*\*\*\*\*”的部分程序代码，可以发现该代码具有以下的优点：

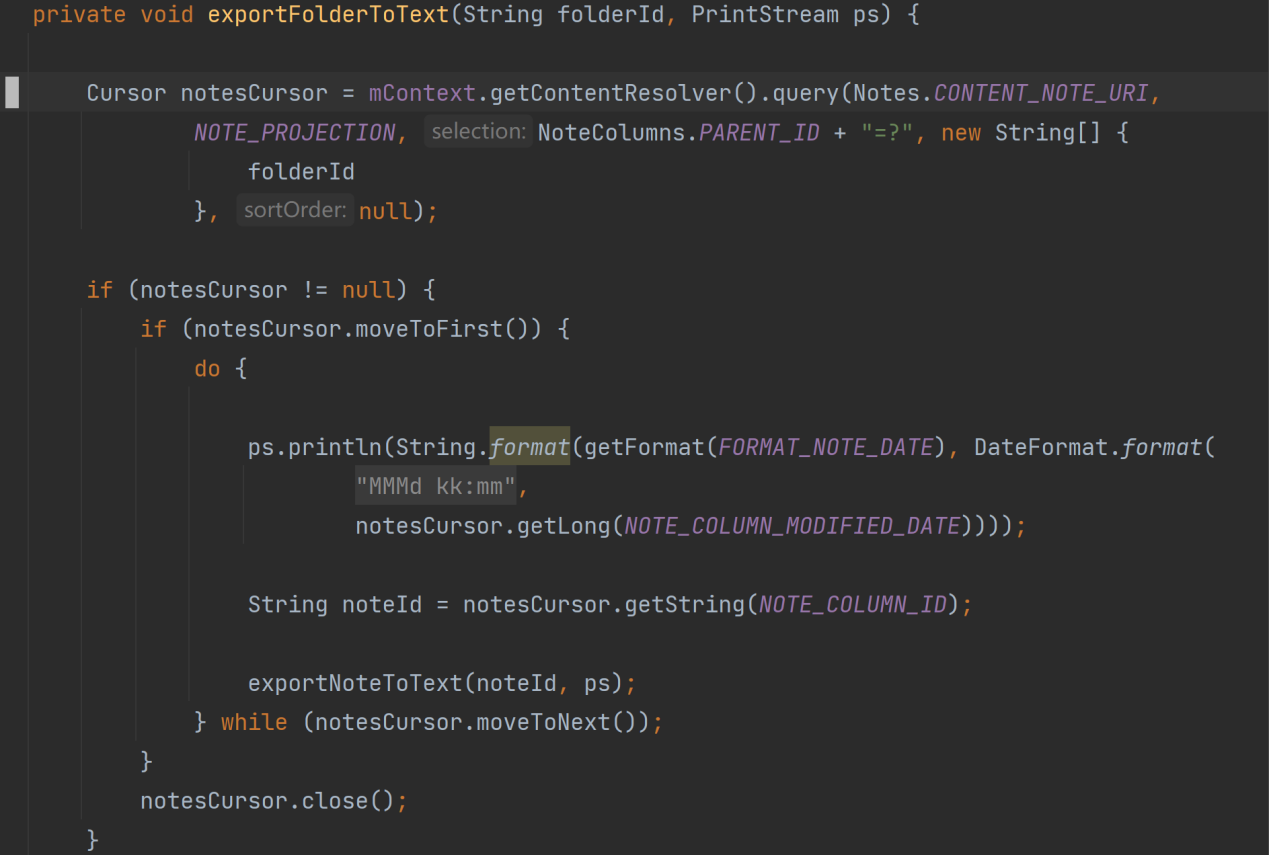
* 较好地遵循了Java编程规范，如方法的命名、常元和变元的等。命名(如“TAG”、“folderID”)
* 程序代码具有良好的封装性、可重用性、可扩展性和可维护性，体现了高内聚、低耦合度的模块化设计思想。例如，方法“getNewNoteId0”所封装的功能具有良好的独立性。
* 对可能出现的异常进行了处理。例如在图示的代码中，对“NumberFormatException”异常进行了处理。
* 对方法进行了必要的注释，如提供了“create anew note idforaddinganew note to database”注释以描述该方法的功能，目注释简洁、直观、易于理解。

但是从质量的视角，该开源软件代码尚存在一些美中不足之处，具体表现为：

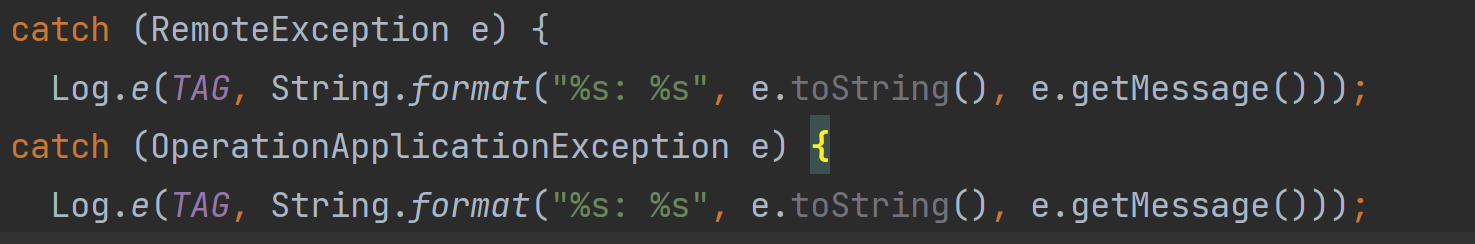
* 关键语句缺乏必要的注释。例如，语句“if(noteId == -1)”缺乏必要的注释以解释此处“-1”表示何意，进而影响对该关键语句的理解，导致后期对该语句的维护会较为困难。这一问题同....get(1)....”语句中，其中的符号“1”表示何意，这样出现在“些程序代码需要提供必要的注释以解释清楚这些符号的内涵。
* 功能实现的方式实现不恰当，容易出现质量问题。例如，BackupUtilsjava”代码中出现了资源对象在使用完之后未被关闭的现象，从而易发生内存泄漏的问题。资源对象的创建应采用“ttry-withresources”的模式，以防止出现上述问题，为此需在finally块中添加“fos.close()”语句。



* 方法内部的实现代码过于复杂，不易理解和维护。例如，在BackupUtils.java代码中有多个嵌套的“If语句”，使得整个程序的控制流嵌套层数太多，导致程序控制结构较为复杂，影响程序代码的可理解性和可维护性。



* 字符串不应该被复制，在代码重构的过程中，重复的字符串如果需要进行修改，则容易发生因为仅有其中某个或某一部分值进行修改的数据一致性错误。例如，在DataUtils.java中有字符串的重复使用



* 实例的方法不能够修改静态域变量的值，否则会在出现多个该类实例和多线程编程问题。例如，在GTaskSyncService.java就出现这样的问题。



* Switch 块需要 default 语句要么进行相应的操作，要么说明为什么不需要采取任何行动。例如，在NoteEditText.java就出现这样的问题。



* 工具类不应该含有公有构造器。工具类中，只含有静态的方法，可以使用类直接调用而不需要实例化对象，而使用私有构造器的目的是在类的外部无法调用该构造器，从而能够阻止对象的生成;Java 在未显式声明私有构造器的时候默认添加一个公有构造器，可能会使该工具类出现未知的错误。

### 使用SonarQube进行自动分析

一般地，根据严重程度的差异性，可以将程序代码中的缺陷和问题分为以下几个不同的等级。

* Blocker：严重程度最高，极有可能造成系统和应用程序崩溃和功能丧失，比如死循环问题。
* Critical：严重程度较高，可能存在影响程序运行的错误或者安全缺陷。
* Major：严重程度一般，如存在部分次要功能没有完全实现。
* Minor：严重程度较低，在一定程度上给用户带来不便。
* Info：严重程度最低，大多为代码质量分析软件提出的一些改进和建议。

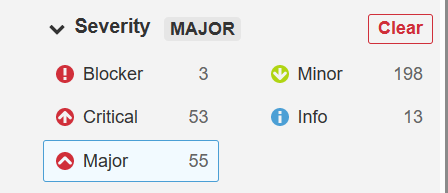
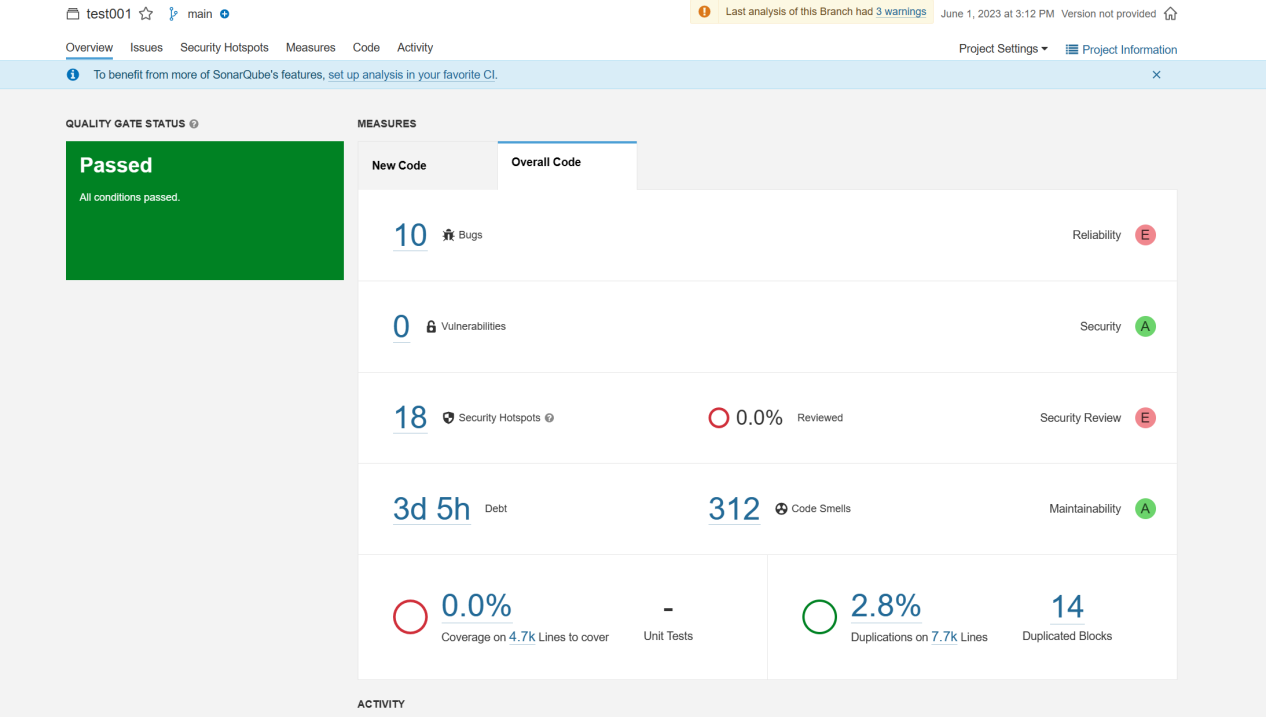
在Android Studio中使用SonarQube对所加载程序进行分析。通过分析，SonarQube工具反馈有关“小米便签”开源软件代码的质量报告如下： 

图1 SonarQube分析结果

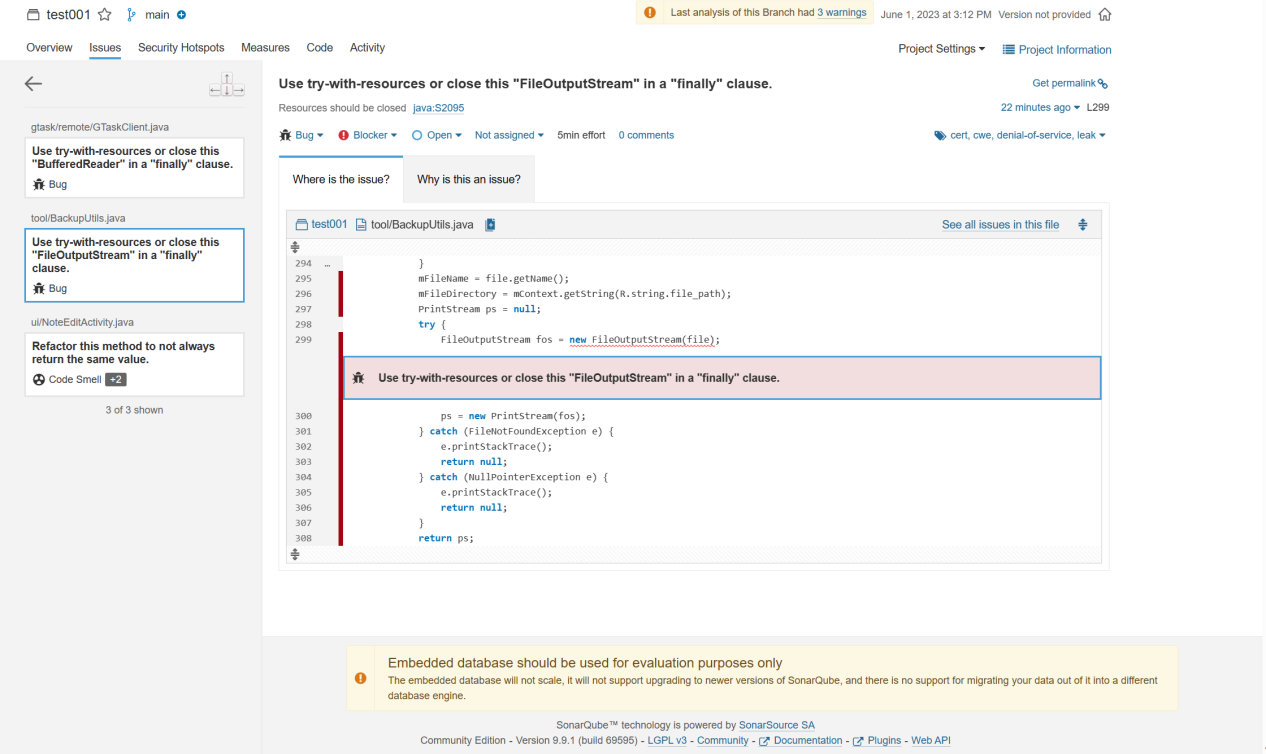
在小米便签软件的源代码中，共审查出42个文件中的375个问题。

现对代码存在的质量问题进行汇总统计，对严重程度为Blocker、Critical和Major的问题进行逐一的分析：

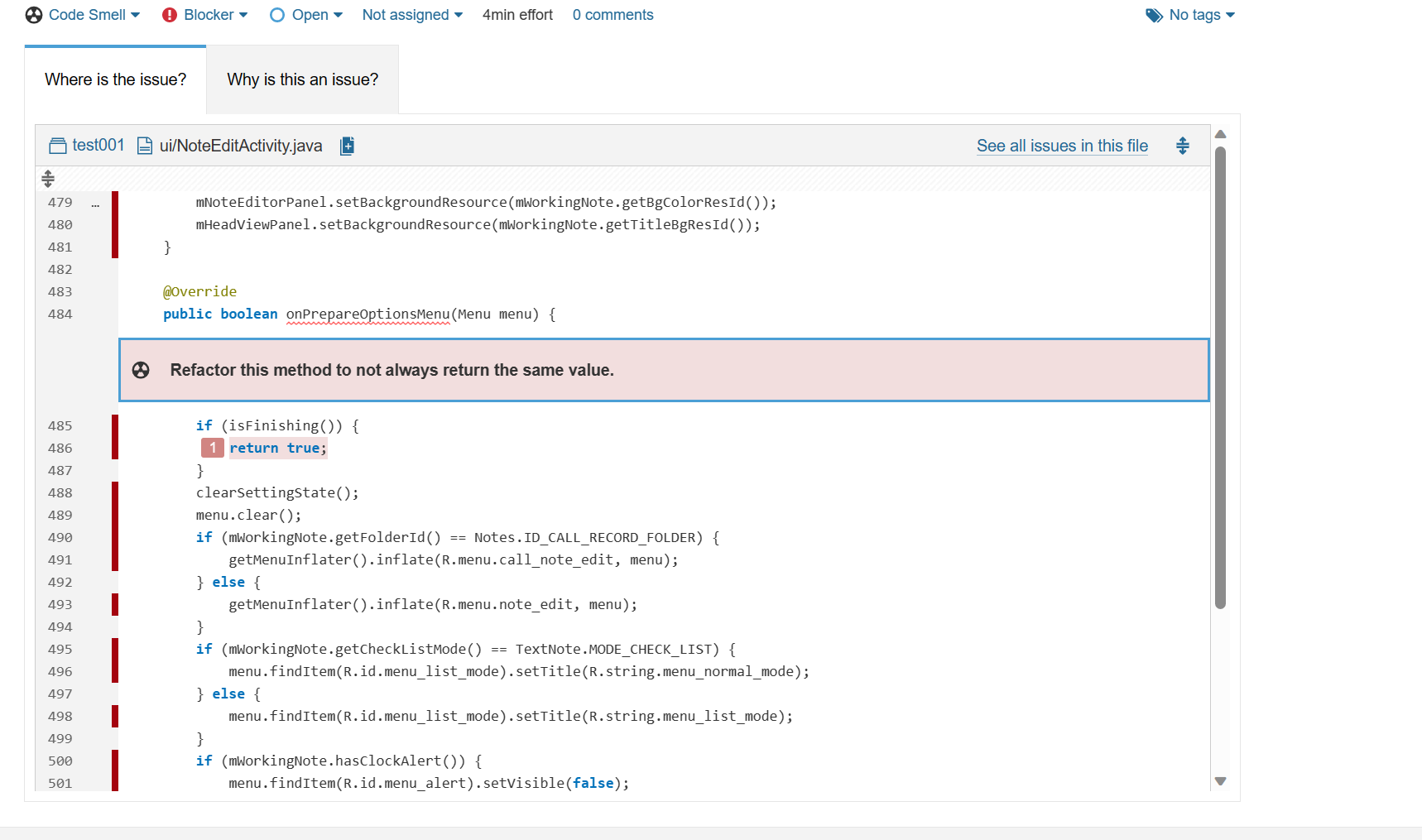
1. **严重程度：Blocker**
2. 应该在“BuffereReader”这个里面加一个“finally”模块



1. 应该在“FileOutStream”这个里面加一个“finally”字块



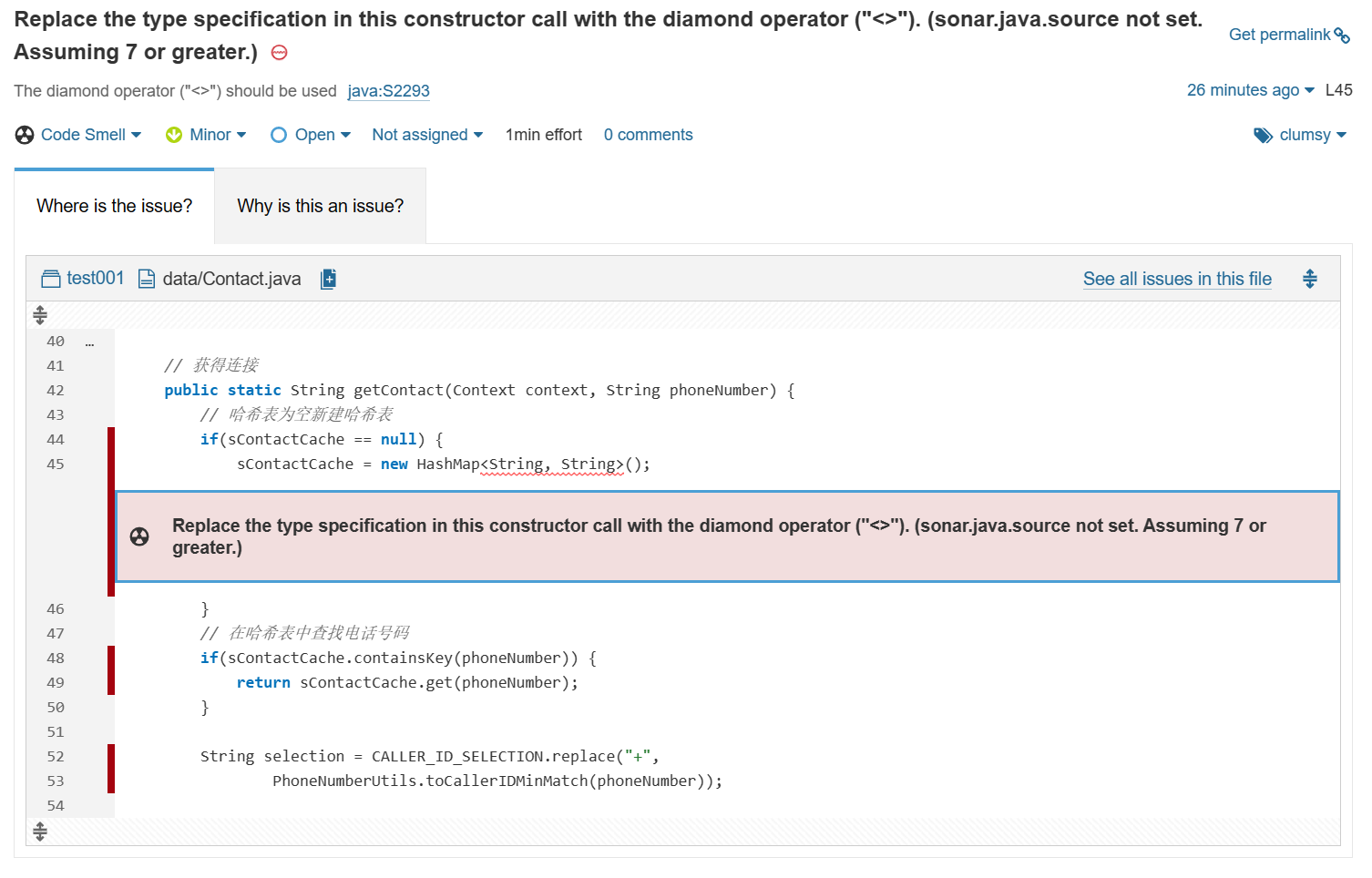
1. 应该在第一个if语句里面增加一个返回量true



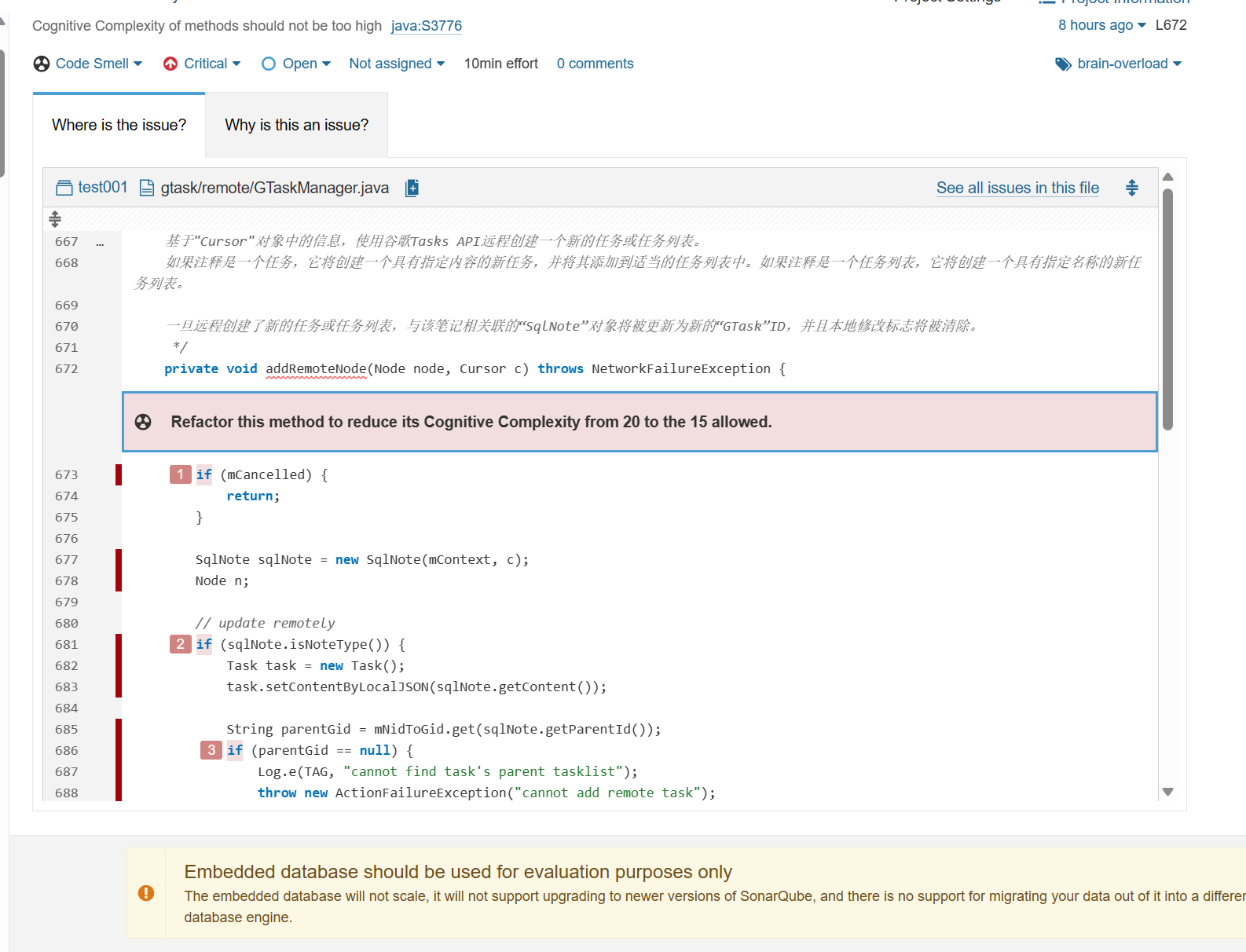
1. **严重程度：Critical**
2. 定义一个private函数



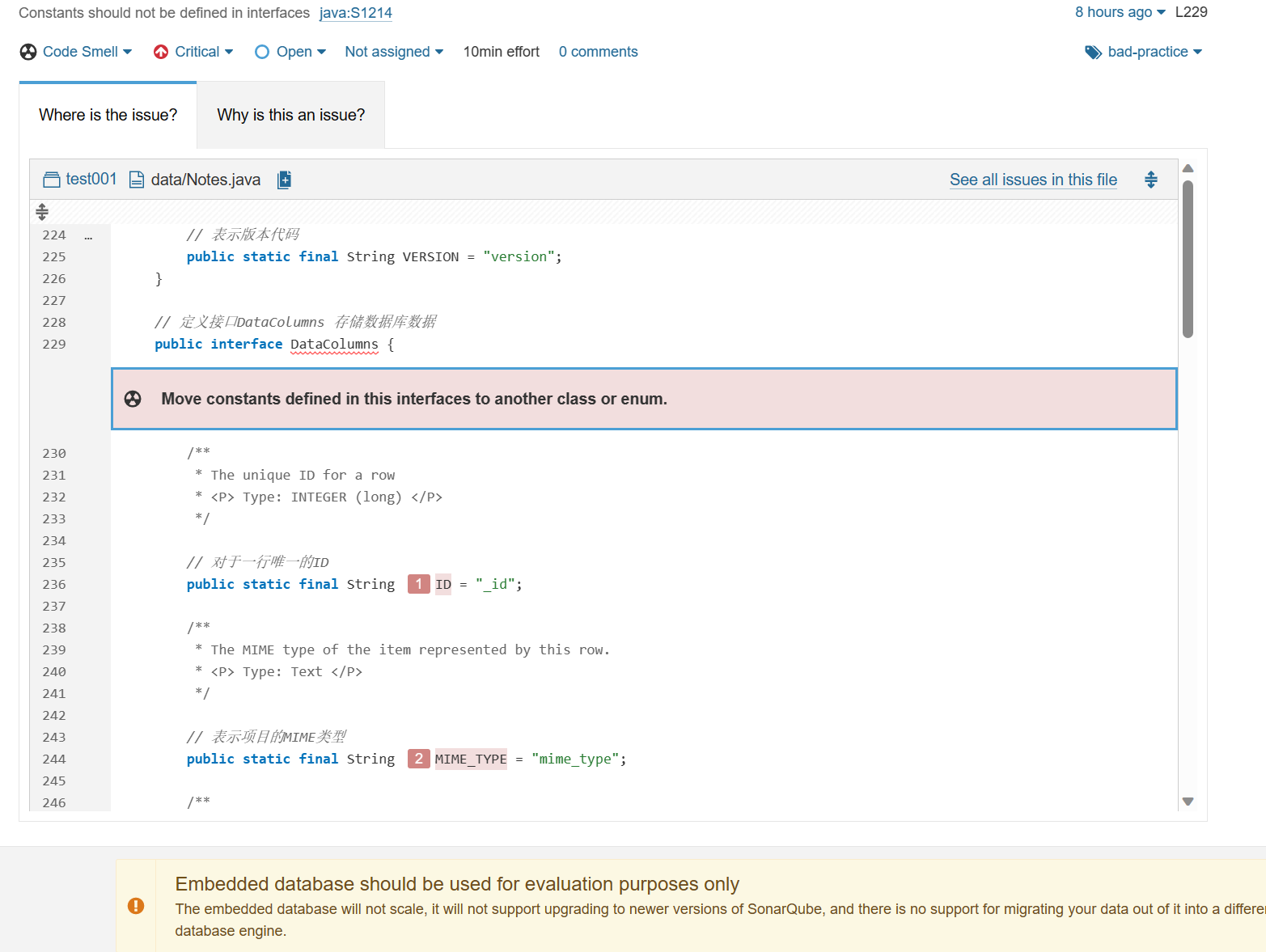
1. 用“<>”来替换原来的“<String,String>”



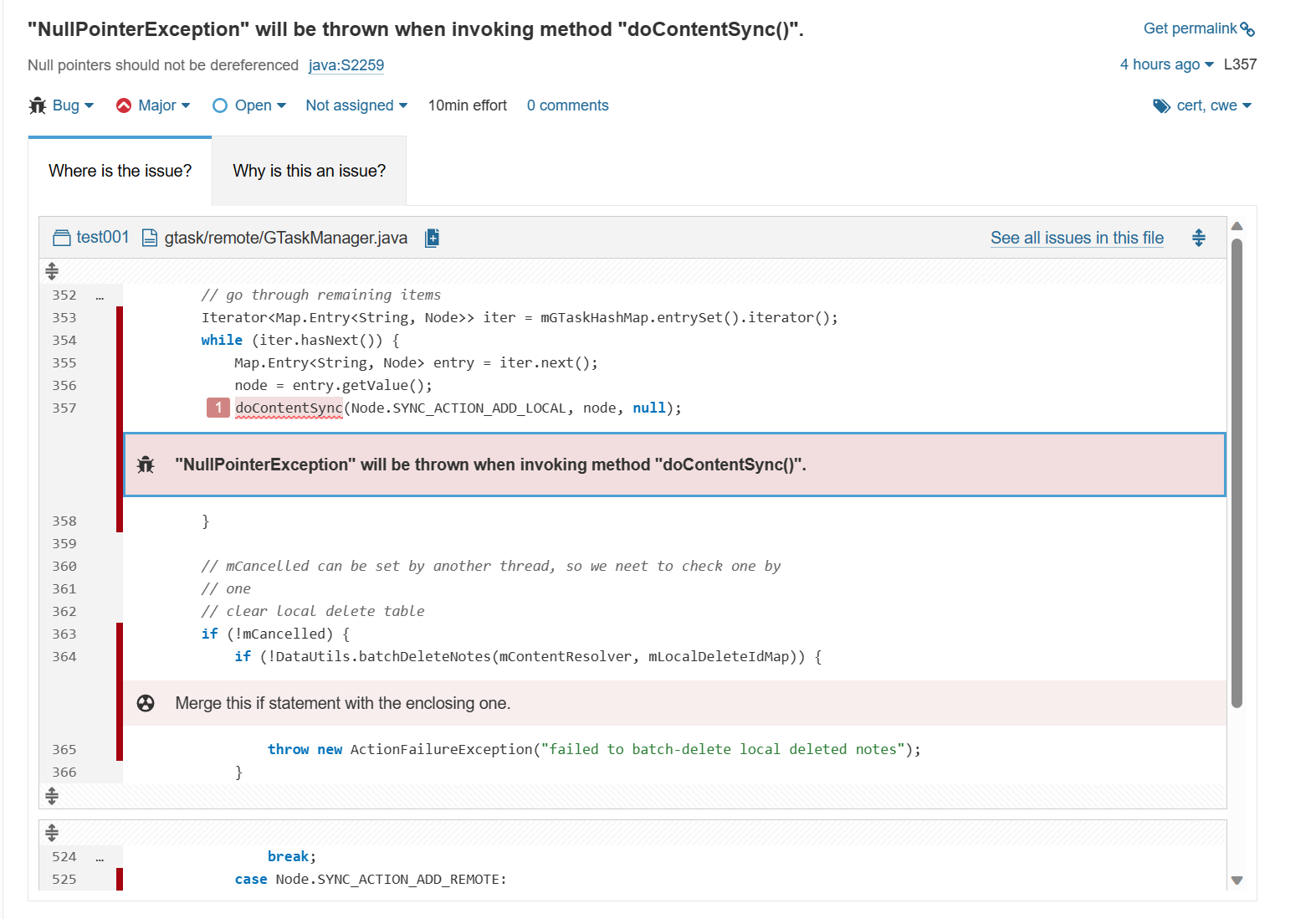
1. 减少此方法的if-else嵌套程度



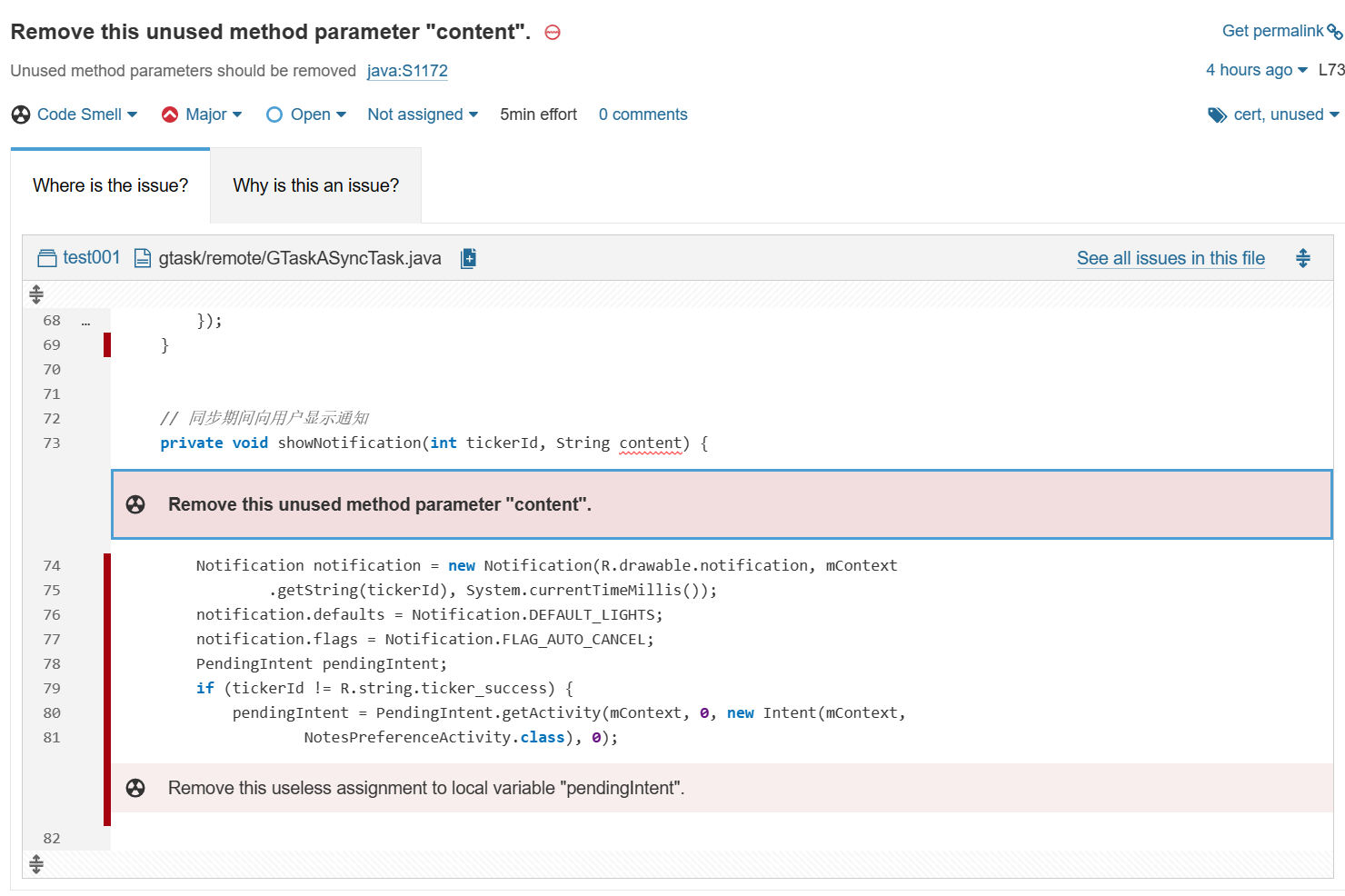
1. 将此接口中定义的常量移动到另一个类或枚举

\

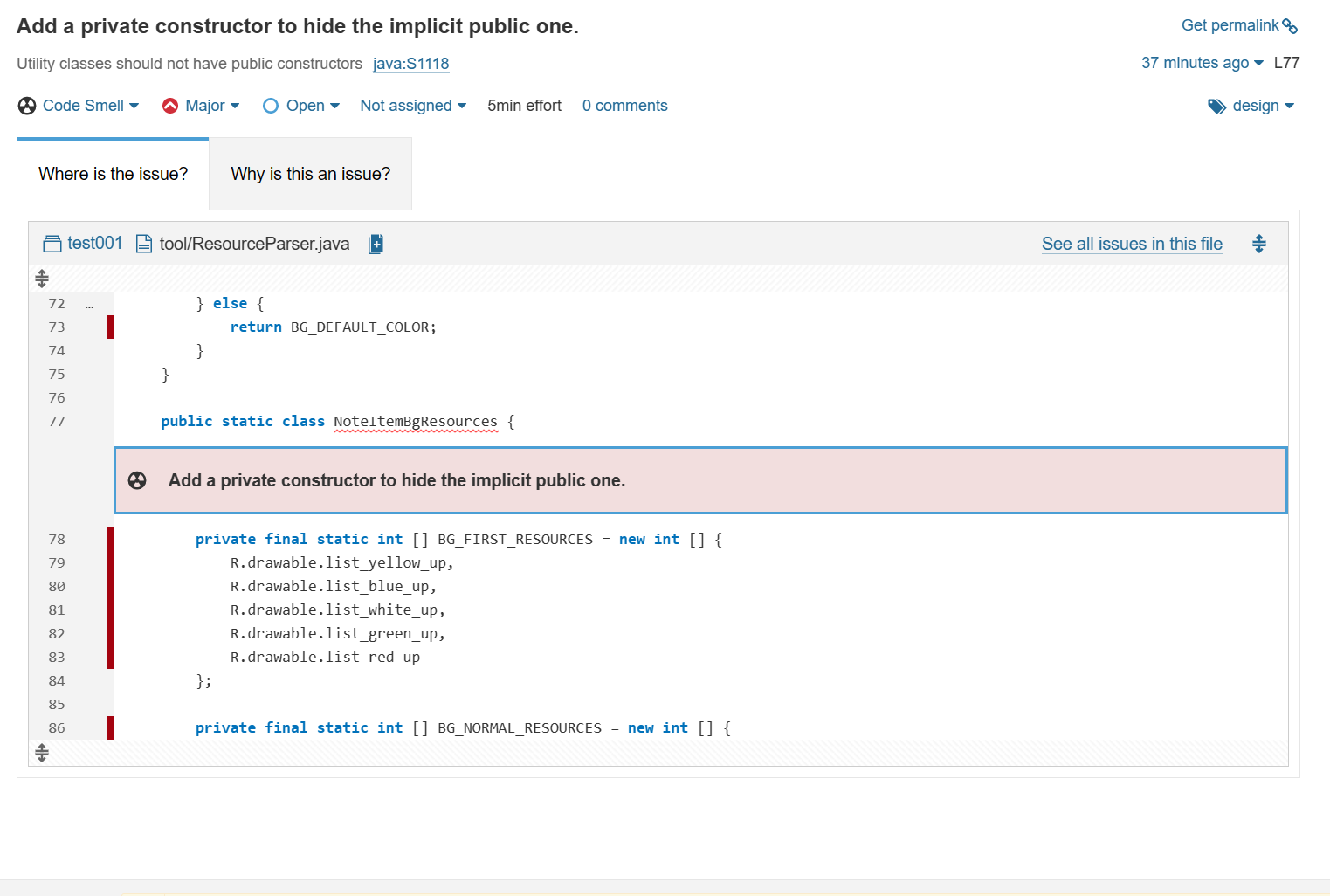
1. **严重程度：Major**
2. 需要在方法里面添加一个判断非NULL的情况



1. 移除多余参数content



1. 增加一个私有变量来隐藏公有变量



1. 使用entrySet代替keySet

