Security

Hotspot

(36)

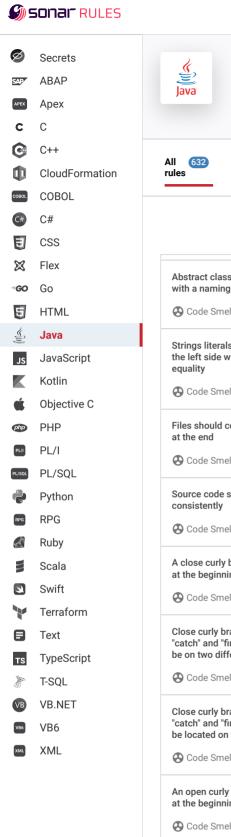
Java static code analysis

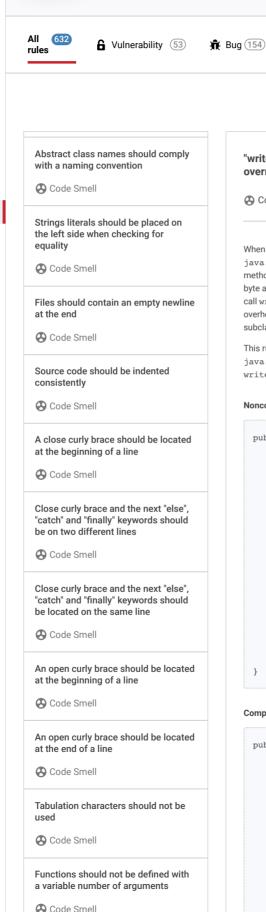
Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your JAVA code

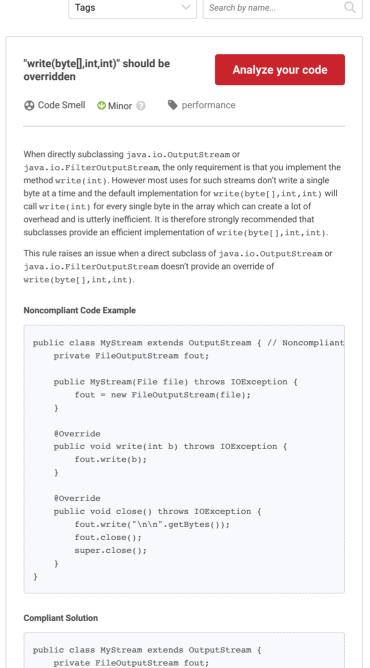


Products >

O Quick 42







public MyStream(File file) throws IOException { fout = new FileOutputStream(file);

public void write(int b) throws IOException {

@Override

@Override

fout.write(b);

⊗ Code

Smell

(389)

Local-Variable Type Inference should be used

Code Smell

Migrate your tests from JUnit4 to the new JUnit5 annotations

Code Smell

Track uses of disallowed classes

Code Smell

Track uses of "@SuppressWarnings" annotations

Code Smell

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.

Privacy Policy