sonar RULES

Products ⌄

Secrets
ABAP
Apex
C
C++
CloudFormation
COBOL
C#
CSS
Flex
Go
HTML
Java
JavaScript
Kotlin
Objective C
PHP
PL/I
PL/SQL
Python
RPG
Ruby
Scala
Swift
Terraform
Text
TypeScript
T-SQL
VB.NET
VB6
XML

Java static code analysis

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

| All rules 632 | 🔒 Vulnerability 53 | 🐛 Bug 154 | 🛡 Security Hotspot 36 | Code Smell 389 | Quick Fix 42 |

Tags ⌄   Search by name...

Abstract class names should comply with a naming convention

⚙ Code Smell

Strings literals should be placed on the left side when checking for equality

⚙ Code Smell

Files should contain an empty newline at the end

⚙ Code Smell

Source code should be indented consistently

⚙ Code Smell

A close curly brace should be located at the beginning of a line

⚙ Code Smell

Close curly brace and the next "else", "catch" and "finally" keywords should be on two different lines

⚙ Code Smell

Close curly brace and the next "else", "catch" and "finally" keywords should be located on the same line

⚙ Code Smell

An open curly brace should be located at the beginning of a line

⚙ Code Smell

An open curly brace should be located at the end of a line

⚙ Code Smell

Tabulation characters should not be used

⚙ Code Smell

Functions should not be defined with a variable number of arguments

⚙ Code Smell

Methods should not have too many return statements

Analyze your code

⚙ Code Smell   ⬦ Major ?   🏷 brain-overload

Having too many return statements in a method increases the method's essential complexity because the flow of execution is broken each time a return statement is encountered. This makes it harder to read and understand the logic of the method.

Noncompliant Code Example

With the default threshold of 3:

```java
public boolean myMethod() { // Noncompliant; there are 4 ret
  if (condition1) {
    return true;
  } else {
    if (condition2) {
      return false;
    } else {
      return true;
    }
  }
  return false;
}
```

Available In:

sonarlint ⊖ | sonarcloud ⬡ | sonarqube 〜

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