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...

Code Smell

Collection.isEmpty() should be used to test for emptiness
Code Smell

String.valueOf() should not be appended to a String
Code Smell

Interface names should comply with a naming convention
Code Smell

"throws" declarations should not be superfluous
Code Smell

Unnecessary imports should be removed
Code Smell

Return of boolean expressions should not be wrapped into an "if-then-else" statement
Code Smell

Boolean literals should not be redundant
Code Smell

Modifiers should be declared in the correct order
Code Smell

Empty statements should be removed
Code Smell

Class variable fields should not have public accessibility
Code Smell

URIs should not be hardcoded
Code Smell

AssertJ "assertThatThrownBy" should not be used alone

Analyze your code

Code Smell Major tests assertj

Unlike similar AssertJ methods testing exceptions (`assertThatCode()`, `assertThatExceptionOfType()`, ...), the `assertThatThrownBy()` method can be used alone, failing if the code did not raise any exception.

Still, only testing that an exception was raised is not enough to guarantee that it was the expected one, and you should test the exception type or content further. In addition, it will make explicit what you are expecting, without relying on side-effects.

This rule raises an issue when `assertThatThrownBy` is used, without testing the exception further.

Noncompliant Code Example

```
assertThatThrownBy(() -> shouldThrow()); // Noncompliant, is
```

Compliant Solution

```
assertThatThrownBy(() -> shouldThrow()).isInstanceOf(IOExcep
//or
assertThatThrownBy(() -> shouldThrow()).hasMessage("My excep
```

Available In:

sonarlint sonarcloud sonarqube

Class names should comply with a naming convention

Code Smell

Method names should comply with a naming convention

Code Smell

Comma-separated labels should be used in Switch with colon case

Code Smell

JUnit5 test classes and methods should have default package visibility

Code Smell