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

Scala static code analysis

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

All rules 41 Bug 6 Security Hotspot 2 Code Smell 33

Tags

Search by name...

Code Smell

Methods should not be empty
Code Smell

All branches in a conditional structure should not have exactly the same implementation
Bug

Related "if"/"else if" statements and "case" in a "match" should not have the same condition
Bug

Identical expressions should not be used on both sides of a binary operator
Bug

All code should be reachable
Bug

Variables should not be self-assigned
Bug

Useless "if(true) {...}" and "if(false){...}" blocks should be removed
Bug

Methods should not have identical implementations
Code Smell

Two branches in a conditional structure should not have exactly the same implementation
Code Smell

"match" expressions should not have too many "case" clauses
Code Smell

Sections of code should not be commented out

Cognitive Complexity of functions should not be too high

Analyze your code

Code Smell Critical brain-overload

Cognitive Complexity is a measure of how hard the control flow of a function is to understand. Functions with high Cognitive Complexity will be difficult to maintain.

See

- Cognitive Complexity

Available In:

sonarlint sonarcloud sonarqube

commented out

Code Smell

Unused function parameters should be removed

Code Smell

Unused "private" methods should be removed

Code Smell

Track uses of "FIXME" tags

Code Smell

Nested blocks of code should not be left empty

Code Smell