




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... 🔍

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

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

Functions should not have too many parameters

Code Smell

Collapsible "if" statements should be merged

Code Smell

Using hardcoded IP addresses is security-sensitive

Security Hotspot

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

Analyze your code

Bug

Major

Using the same value on either side of a binary operator is almost always a mistake. In the case of logical operators, it is either a copy/paste error and therefore a bug, or it is simply wasted code, and should be simplified. In the case of bitwise operators and most binary mathematical operators, having the same value on both sides of an operator yields predictable results, and should be simplified.

Exceptions

This rule ignores *, +, and =.

See

- {rule:scala:S1656} - Implements a check on =.

Available In:

sonarlint





sonarcloud

sonarqube

© 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](#)

https://rules.sonarsource.com/scala/RSPEC-1764

1/2

<div>Multi-line comments should not be empty</div> <div> Code Smell</div>
<div>Boolean checks should not be inverted</div> <div> Code Smell</div>
<div>Unused local variables should be removed</div> <div> Code Smell</div>
<div>Local variable and function parameter names should comply with a naming convention</div> <div> Code Smell</div>
<div>Boolean literals should not be</div>