




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... 🔍


Bug


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

Bug


Bug


All code should be reachable

Bug


Bug


Variables should not be self-assigned

Bug


Bug


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

Bug


Code Smell


Methods should not have identical implementations

Code Smell


Code Smell

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

Code Smell

Code Smell

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

Code Smell

Code Smell

Sections of code should not be commented out

Code Smell

Code Smell

Unused function parameters should be removed

Code Smell

Code Smell

Unused "private" methods should be removed

Code Smell


Code Smell


Track uses of "FIXME" tags


Code Smell

Methods should not be empty

Analyze your code

Code Smell

Critical ?

suspicious

There are several reasons for a method not to have a method body:

- It is an unintentional omission, and should be fixed to prevent an unexpected behavior in production.
- It is not yet, or never will be, supported. In this case an exception should be thrown.
- The method is an intentionally-blank override. In this case a nested comment should explain the reason for the blank override.

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-1186

1/2

| |
|---|
| <div><div>Nested blocks of code should not be left empty</div><div> Code Smell</div></div> |
| <div><div>Functions should not have too many parameters</div><div> Code Smell</div></div> |
| <div><div>Collapsible "if" statements should be merged</div><div> Code Smell</div></div> |
| <div><div>Using hardcoded IP addresses is security-sensitive</div><div> Security Hotspot</div></div> |
| <div><div>Multi-line comments should not be</div></div> |