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

Local variable and function parameter names should comply with a naming convention

☢ Code Smell

Boolean literals should not be redundant

☢ Code Smell

Class names should comply with a naming convention

☢ Code Smell

Method names should comply with a naming convention

☢ Code Smell

Track uses of "TODO" tags

☢ Code Smell

Track lack of copyright and license headers

☢ Code Smell

"match" statements should not be nested

☢ Code Smell

Control flow statements "if", "for", "while", "match" and "try" should not be nested too deeply

☢ Code Smell

"if ... else if" constructs should end with "else" clauses

☢ Code Smell

Expressions should not be too complex

☢ Code Smell

Scala parser failure

☢ Code Smell

Sections of code should not be commented out

Analyze your code

☢ Code Smell   ⬥ Major ⍰   🏷 unused

Programmers should not comment out code as it bloats programs and reduces readability.

Unused code should be deleted and can be retrieved from source control history if required.

Available In:

sonarlint 😊 | sonarcloud ⬡ | sonarqube 〰

Code Smell

Methods should not have too many lines of code

Code Smell

Statements should be on separate lines

Code Smell

"match case" clauses should not have too many lines of code

Code Smell

Files should not have too many lines of code

Code Smell