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...

"Arrays.stream" should be used for primitive arrays
Code Smell

Printf-style format strings should be used correctly
Code Smell

Assertion arguments should be passed in the correct order
Code Smell

Ternary operators should not be nested
Code Smell

"writeObject" should not be the only "synchronized" code in a class
Code Smell

Reflection should not be used to increase accessibility of classes, methods, or fields
Code Smell

Static fields should not be updated in constructors
Code Smell

"Thread.sleep" should not be used in tests
Code Smell

"entrySet()" should be iterated when both the key and value are needed
Code Smell

"DateUtils.truncate" from Apache Commons Lang library should not be used
Code Smell

Multiline blocks should be enclosed in curly braces
Code Smell

Regex alternatives should not be redundant

Analyze your code

Bug Major regex

If an alternative in a regular expression only matches things that are already matched by another alternative, that alternative is redundant and serves no purpose.

In the best case this means that the offending subpattern is merely redundant and should be removed. In the worst case it’s a sign that this regex does not match what it was intended to match and should be reworked.

Noncompliant Code Example

```
"[ab]|a"    // The "|a" is redundant because "[ab]" already m
".*|a"      // .* matches everything, so any other alternativ
```

Compliant Solution

```
"[ab]"
".*"
```

Available In:

sonarlint sonarcloud sonarqube

"readObject" should not be "synchronized"

Code Smell

"Preconditions" and logging arguments should not require evaluation

Code Smell

Boolean expressions should not be gratuitous

Code Smell

"Lock" objects should not be "synchronized"

Code Smell