sonar RULES

Products ⌄

⊘	Secrets
SAP	ABAP
APEX	Apex
C	C
C++	C++
☁	CloudFormation
COBOL	COBOL
C#	C#
CSS	CSS
✕	Flex
⟶GO	Go
HTML	HTML
☕	**Java**
JS	JavaScript
Kotlin	Kotlin
🍎	Objective C
php	PHP
PL/I	PL/I
PL/SQL	PL/SQL
Python	Python
RPG	RPG
Ruby	Ruby
Scala	Scala
Swift	Swift
Terraform	Terraform
Text	Text
TS	TypeScript
T-SQL	T-SQL
VB	VB.NET
VB6	VB6
XML	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... 🔍

🔒 Vulnerability

"ActiveMQConnectionFactory" should not be vulnerable to malicious code deserialization

🔒 Vulnerability

Logging should not be vulnerable to injection attacks

🔒 Vulnerability

Exceptions should not be thrown from servlet methods

🔒 Vulnerability

Return values should not be ignored when they contain the operation status code

🐛 Bug

Repeated patterns in regular expressions should not match the empty string

🐛 Bug

AssertJ assertions "allMatch" and "doesNotContains" should also test for emptiness

🐛 Bug

Double Brace Initialization should not be used

🐛 Bug

Non-primitive fields should not be "volatile"

🐛 Bug

"toArray" should be passed an array of the proper type

🐛 Bug

Neither "Math.abs" nor negation should be used on numbers that could be "MIN_VALUE"

🐛 Bug

"wait", "notify" and "notifyAll" should only be called when a lock is obviously held on an object

Analyze your code

🐛 Bug 🔺 Major ⑦ 🏷 multi-threading

By contract, the method `Object.wait(...)`, `Object.notify()` and `Object.notifyAll()` should be called by a thread that is the owner of the object's monitor. If this is not the case an `IllegalMonitorStateException` exception is thrown. This rule reinforces this constraint by making it mandatory to call one of these methods only inside a `synchronized` method or statement.

Noncompliant Code Example

```
private void removeElement() {
  while (!suitableCondition()){
    obj.wait();
  }
  ... // Perform removal
}
```

or

```
private void removeElement() {
  while (!suitableCondition()){
    wait();
  }
  ... // Perform removal
}
```

Compliant Solution

```
private void removeElement() {
  synchronized(obj) {
    while (!suitableCondition()){
      obj.wait();
    }
    ... // Perform removal
  }
}
```

or

```
private synchronized void removeElement() {
  while (!suitableCondition()){
    wait();
  }
  ... // Perform removal
}
```
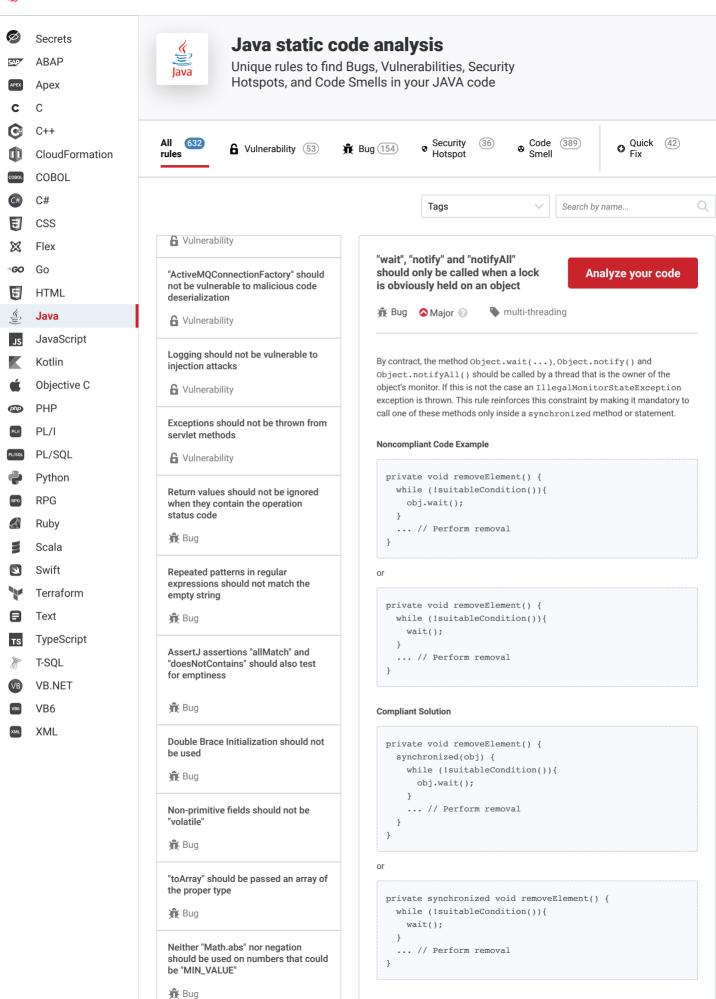
Available In:

The value returned from a stream read should be checked

🐞 Bug

"@NonNull" values should not be set to null

🐞 Bug

"Iterator.next()" methods should throw "NoSuchElementException"

🐞 Bug

"compareTo" results should not be checked for specific values

🐞 Bug

sonarlint 😶 | sonarcloud 🔆 | sonarqube 〵