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...

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

Code Smell

Multiline blocks should be enclosed in curly braces

Code Smell

"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

Classes with only "static" methods should not be instantiated

Code Smell

"Threads" should not be used where "Runnables" are expected

Code Smell

Inner class calls to super class methods should be unambiguous

Code Smell

Unused type parameters should be removed

Code Smell

Parameters should be passed in the correct order

Code Smell

InputSteam.read() implementation should not return a signed byte

Analyze your code

Bug Major

According to the Java documentation, any implementation of the `InputSteam.read()` method is supposed to read the next byte of data from the input stream. The value byte must be an `int` in the range 0 to 255. If no byte is available because the end of the stream has been reached, the value -1 is returned.

But in Java, the `byte` primitive data type is an 8-bit signed two's complement integer. It has a minimum value of -128 and a maximum value of 127. So by contract, the implementation of an `InputSteam.read()` method should never directly return a `byte` primitive data type. A conversion into an unsigned byte must be done before by applying a bitmask.

Noncompliant Code Example

```
@Override
public int read() throws IOException {
  if (pos == buffer.length()) {
    return -1;
  }
  return buffer.getByte(pos++); // Noncompliant, a signed by
}
```

Compliant Solution

```
@Override
public int read() throws IOException {
  if (pos == buffer.length()) {
    return -1;
  }
  return buffer.getByte(pos++) & 0xFF; // The 0xFF bitmask i
}
```

Available In:

sonarlint | sonarcloud | sonarqube

"ResultSet.isLast()" should not be used

Code Smell

"static" members should be accessed statically

Code Smell

Silly math should not be performed

Code Smell

Classes named like "Exception" should extend "Exception" or a subclass

Code Smell