

































-  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 rules632

Vulnerability53

Bug154

Security Hotspot36

Code Smell389

Quick Fix42

Tags ▾

Search by name... 🔍

Classes should not be coupled to too many other classes (Single Responsibility Principle)

Code Smell

"java.lang.Error" should not be extended

Code Smell

Anonymous classes should not have too many lines

Code Smell

Public types, methods and fields (API) should be documented with Javadoc

Code Smell

Exception handlers should preserve the original exceptions

Code Smell

Checked exceptions should not be thrown

Code Smell

Public methods should throw at most one checked exception

Code Smell

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

Code Smell

Methods should not have too many return statements

Code Smell

Magic numbers should not be used

Code Smell

Files should not have too many lines of code

Code Smell

Lines should not be too long

"ResultSet.isLast()" should not be used

Analyze your code

Code Smell

Major ?

performance pitfall sql

There are several reasons to avoid `ResultSet.isLast()`. First, support for this method is optional for `TYPE_FORWARD_ONLY` result sets. Second, it can be expensive (the driver may need to fetch the next row to answer the question). Finally, the specification is not clear on what should be returned when the `ResultSet` is empty, so some drivers may return the opposite of what is expected.

Noncompliant Code Example

```
stmt.executeQuery("SELECT name, address FROM PERSON");
ResultSet rs = stmt.getResultSet();
while (! rs.isLast()) { // Noncompliant
    // process row
}
```

Compliant Solution

```
ResultSet rs = stmt.executeQuery("SELECT name, address FROM
while (rs.next()) {
    // process row
}
```





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](#)

 Code Smell
JEE applications should not "getClassLoader"  Bug
Math should not be performed on floats  Bug
"equals" methods should be symmetric and work for subclasses  Bug
Literal suffixes should be upper case 