

































-  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... 🔍

 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

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

Null pointers should not be dereferenced

Analyze your code

 Bug

 Major

 cwe cert

A reference to null should never be dereferenced/accessed. Doing so will cause a `NullPointerException` to be thrown. At best, such an exception will cause abrupt program termination. At worst, it could expose debugging information that would be useful to an attacker, or it could allow an attacker to bypass security measures.

Note that when they are present, this rule takes advantage of `@CheckForNull` and `@NonNull` annotations defined in [JSR-305](#) to understand which values are and are not nullable except when `@NonNull` is used on the parameter to `equals`, which by contract should always work with null.

Noncompliant Code Example

```
@CheckForNull
String getName(){...}

public boolean isEmpty() {
    return getName().length() == 0; // Noncompliant; the result is null
}
```

```
Connection conn = null;
Statement stmt = null;
try{
    conn = DriverManager.getConnection(DB_URL,USER,PASS);
    stmt = conn.createStatement();
    // ...







}catch(Exception e){
    e.printStackTrace();
}finally{
    stmt.close(); // Noncompliant; stmt could be null if an exception is thrown
    conn.close(); // Noncompliant; conn could be null if an exception is thrown
}
```

```
private void merge(@NonNull Color firstColor, @NonNull Color secondColor) {
    public void append(@CheckForNull Color color) {
        merge(currentColor, color); // Noncompliant; color should not be null
    }
}
```

```
void paint(Color color) {
    if(color == null) {
        System.out.println("Unable to apply color " + color.toString());
        return;
    }
    // ...
}
```

https://rules.sonarsource.com/java/RSPEC-2259

1/2

<div><div>NoSuchElementException</div><div> Bug</div></div>	<div><div>See</div><div><ul style="list-style-type: none">• MITRE, CWE-476 - NULL Pointer Dereference• CERT, EXP34-C. - Do not dereference null pointers• CERT, EXP01-J. - Do not use a null in a case where an object is required</div><div>Available In:</div><div><div>sonarlint</div><div></div><div>sonarcloud</div><div></div><div>sonarqube</div></div></div>
<div><div>"compareTo" results should not be checked for specific values</div><div> Bug</div></div>	
<div><div>Math operands should be cast before assignment</div><div> Bug</div></div>	
<div><div>Ints and longs should not be shifted by zero or more than their number of bits-1</div><div> Bug</div></div>	

© 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](#)