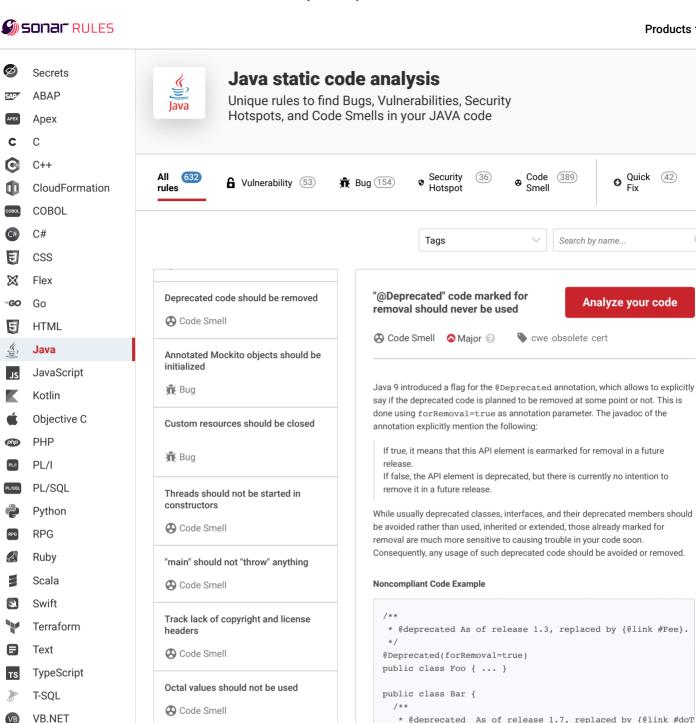


Products ✓

O Quick 42



Exit methods should not be called

HTTP response headers should not be vulnerable to injection attacks

Members of Spring components

Classes should not be loaded

Equality operators should not be used in "for" loop termination conditions

Code Smell

Vulnerability

should be injected

Vulnerability

Vulnerability

Code Smell

dynamically

```
* @deprecated As of release 1.3, replaced by {@link #Fee}.
   * @deprecated As of release 1.7, replaced by {@link #doT
  @Deprecated(forRemoval=true)
  public void doTheThing() { ... }
  public void doTheThingBetter() { ... }
   * @deprecated As of release 1.14 due to poor performances
  @Deprecated(forRemoval=false)
  public void doTheOtherThing() { ... }
public class Qix extends Bar {
  @Override
  public void doTheThing() { \dots } // Noncompliant; don't ov
public class Bar extends Foo { // Noncompliant; Foo is depr
  public void myMethod() {
    Bar bar = new Bar(); // okay; the class isn't deprecate
    bar.doTheThing(); // Noncompliant; doTheThing method is
```

bar.doTheOtherThing(); // Okay; deprecated, but not mark

VB₆

XML

"Bean Validation" (JSR 380) should be properly configured

Code Smell

Spring beans should be considered by "@ComponentScan"

Code Smell

Number patterns should be regular

Code Smell

Lazy initialization of "static" fields should be "synchronized"

Code Smell

Code Smell

See

• MITRE, CWE-477 - Use of Obsolete Functions
• CERT, MET02-J. - Do not use deprecated or obsolete classes or methods
• RSPEC-1874 for standard deprecation use

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