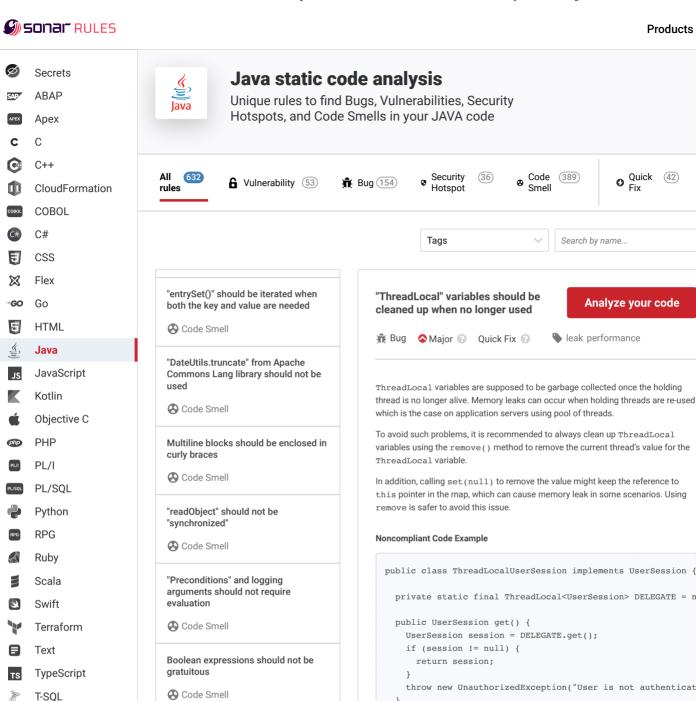
Products >

O Quick 42



"Lock" objects should not be

Classes with only "static" methods should not be instantiated

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

Inner class calls to super class

methods should be unambiguous

Unused type parameters should be

"synchronized"

Code Smell

Code Smell

Code Smell

Code Smell

Code Smell

removed

public class ThreadLocalUserSession implements UserSession { private static final ThreadLocal<UserSession> DELEGATE = n throw new UnauthorizedException("User is not authenticat public void set(UserSession session) { DELEGATE.set(session); public void incorrectCleanup() { DELEGATE.set(null); // Noncompliant // some other methods without a call to DELEGATE.remove()

Compliant Solution

```
public class ThreadLocalUserSession implements UserSession {
  private static final ThreadLocal<UserSession> DELEGATE = n
  public UserSession get() {
    UserSession session = DELEGATE.get();
    if (session != null) {
      return session;
    throw new UnauthorizedException("User is not authenticat
```

VB.NET

VB6

XML

Parameters should be passed in the correct order

Code Smell

"ResultSet.isLast()" should not be

Code Smell

"static" members should be accessed statically

Code Smell

Silly math should not be performed

Code Smell

```
public void set(UserSession session) {
   DELEGATE.set(session);
 public void unload() {
   DELEGATE.remove(); // Compliant
  // ...
}
```

Exceptions

Rule will not detect non-private ThreadLocal variables, because remove() can be called from another class.

• Understanding Memory Leaks in Java

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.

<u>Privacy Policy</u>