Security

Hotspot

**#** Bug (154)

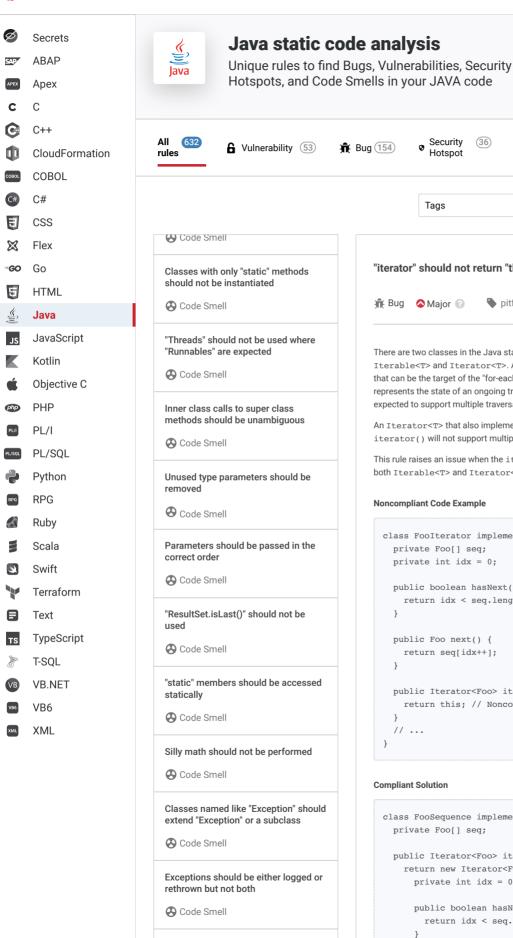
(36)



Products >

O Quick 42 Fix

Q



```
Search by name.
           Tags
"iterator" should not return "this"
                                               Analyze your code
👬 Bug 🔷 Major 🕝
                         pitfall
There are two classes in the Java standard library that deal with iterations:
Iterable<T> and Iterator<T>. An Iterable<T> represents a data structure
that can be the target of the "for-each loop" statement, and an Iterator<T>
represents the state of an ongoing traversal. An Iterable<T> is generally
expected to support multiple traversals.
An Iterator<T> that also implements Iterable<t> by returning itself as its
iterator() will not support multiple traversals since its state will be carried over.
This rule raises an issue when the iterator() method of a class implementing
both Iterable<T> and Iterator<t> returns this.
Noncompliant Code Example
 class FooIterator implements Iterator<Foo>, Iterable<Foo> {
    private Foo[] seg;
    private int idx = 0;
    public boolean hasNext() {
      return idx < seq.length;
    public Foo next() {
      return seq[idx++];
    public Iterator<Foo> iterator() {
      return this; // Noncompliant
Compliant Solution
  class FooSequence implements Iterable<Foo> {
    private Foo[] seq;
    public Iterator<Foo> iterator() {
      return new Iterator<Foo>() {
        private int idx = 0;
        public boolean hasNext() {
```

return idx < seq.length;

public Foo next() { return seq[idx++];

};

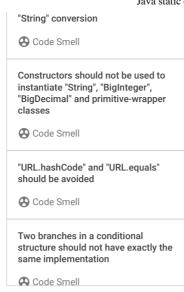
(389)

Objects should not be created only to

Primitives should not be boxed just for

"getClass'

Code Smell





© 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