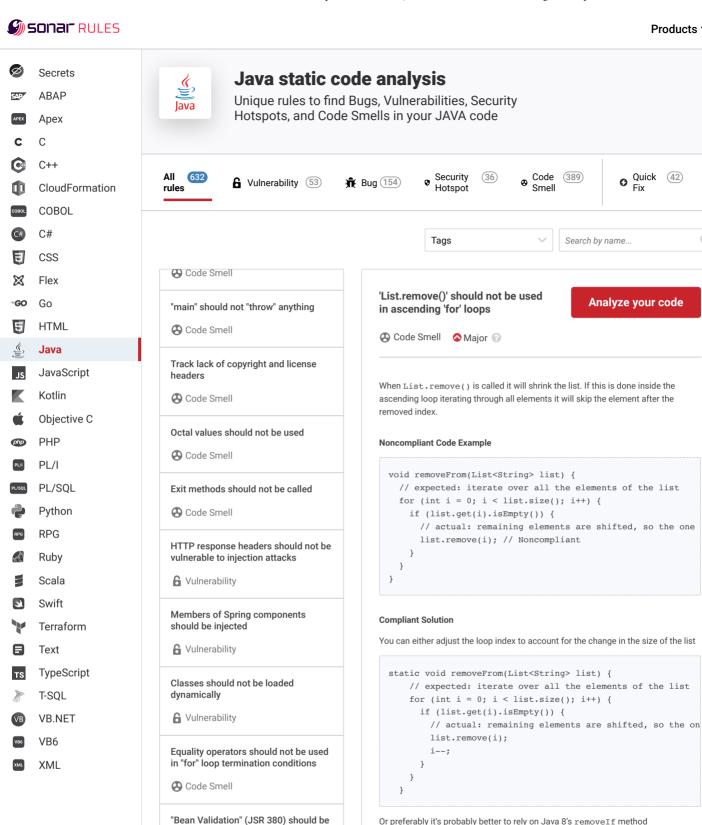
Products ✓

O Quick 42 Fix

(389)

Search by name.



properly configured

"@ComponentScan" Code Smell

Spring beans should be considered by

Number patterns should be regular

Lazy initialization of "static" fields

should be "synchronized"

Code Smell

Code Smell

Code Smell

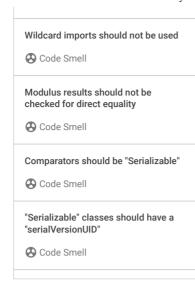
```
Analyze your code
When List.remove() is called it will shrink the list. If this is done inside the
ascending loop iterating through all elements it will skip the element after the
 void removeFrom(List<String> list) {
    \ensuremath{//} expected: iterate over all the elements of the list
    for (int i = 0; i < list.size(); i++) {</pre>
      if (list.get(i).isEmpty()) {
         // actual: remaining elements are shifted, so the one
         list.remove(i); // Noncompliant
You can either adjust the loop index to account for the change in the size of the list
```

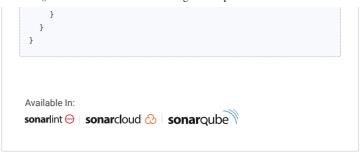
Or preferably it's probably better to rely on Java 8's removeIf method

```
static void removeFrom(List<String> list) {
  list.removeIf(String::isEmpty);
```

The descending loop doesn't have this issue, because the index will be correct when we loop in descending order

```
void removeFrom(List<String> list) {
  for (int i = list.size() - 1; i >= 0; i--) {
    if (list.get(i).isEmpty()) {
      list.remove(i);
```





© 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