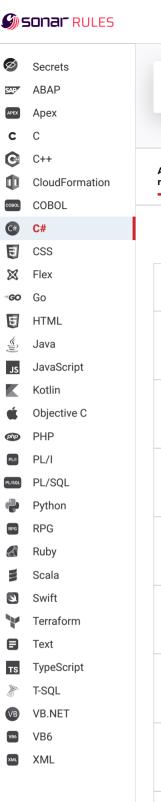
O Quick 52 Fix





```
C# static code analysis
              Unique rules to find Bugs, Vulnerabilities, Security
              Hotspots, and Code Smells in your C# code
ΔII
     409
              6 Vulnerability (34)
                                      # Bug (76)
rules
 "protected" members
 Code Smell
 Underscores should be used to make
 large numbers readable
 Code Smell
 "ToString()" calls should not be
 Code Smell
 "==" should not be used when "Equals"
 is overridden
 Code Smell
 An abstract class should have both
 abstract and concrete methods
 Code Smell
 Multiple variables should not be
 declared on the same line
 Code Smell
 Culture should be specified for "string"
 operations
 Code Smell
 "switch" statements should have at
 least 3 "case" clauses
 Code Smell
 break statements should not be used
 except for switch cases
 Code Smell
 String literals should not be duplicated
 Code Smell
 Files should contain an empty newline
 at the end
 Code Smell
 Unused "using" should be removed
```

Code Smell

```
Q
           Tags
                                        Search by name..
Nested code blocks should not be
                                              Analyze your code
Code Smell Minor
                               bad-practice
Nested code blocks can be used to create a new scope and restrict the visibility of
the variables defined inside it. Using this feature in a method typically indicates that
the method has too many responsibilities, and should be refactored into smaller
methods.
Noncompliant Code Example
 public void Evaluate()
           // Noncompliant - nested code block '{' ... '}'
             int a = stack.pop();
             int b = stack.pop();
             int result = a + b;
             stack.push(result);
      /* ... */
  }
Compliant Solution
 public void Evaluate()
      /* ... */
      StackAdd();
      /* ... */
 private void StackAdd()
        int a = stack.pop();
        int b = stack.pop();
        int result = a + b;
        stack.push(result);
Exceptions
The usage of a code block after a "case" is allowed for this rule.
 Available In:
 sonarlint ⊕ | sonarcloud ↔ | sonarqube
```

**⇔** Code Smell

Security • Secu. Hotspot A close curly brace should be located at the beginning of a line

Code Smell

Tabulation characters should not be used

Code Smell

Methods and properties should be named in PascalCase

Code Smell

Track uses of in-source issue suppressions

Code Smell

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