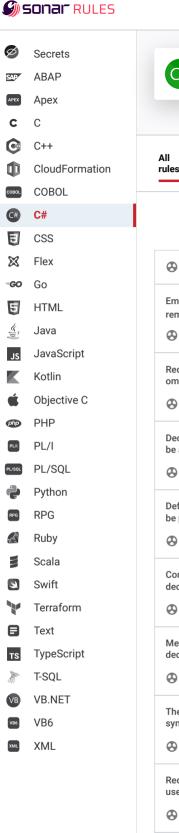
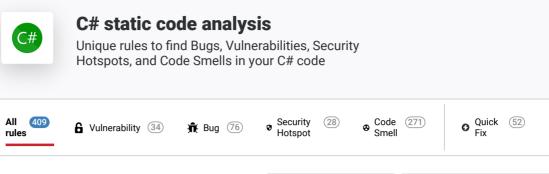
Search by name..

Analyze your code







Tags

"base.Equals" should not be used

to check for reference equality in

"Equals" if "base" is not "object"



Sequential tests should not check the

same condition

Code Smell

```
🙀 Bug 😍 Minor 🔞
object.Equals() overrides can be optimized by checking first for reference
equality between this and the parameter. This check can be implemented by
calling object.ReferenceEquals() or base.Equals(), where base is
object. However, using base. Equals () is a maintenance hazard because while
it works if you extend Object directly, if you introduce a new base class that
overrides Equals, it suddenly stops working.
This rule raises an issue if base. Equals() is used but base is not object.
Noncompliant Code Example
 class Base
    private int baseField;
    public override bool Equals(object other)
      if (base.Equals(other)) // Okay; base is object
         return true;
      return this.baseField == ((Base)other).baseField;
 class Derived : Base
    private int derivedField:
    public override bool Equals(object other)
      if (base.Equals(other)) // Noncompliant
        return true;
      return this.derivedField == ((Derived)other).derivedFiel
  }
```

Compliant Solution

```
class Base
  private int baseField;
  public override bool Equals(object other)
```

Redundant modifiers should not be

Code Smell

Methods and properties that don't access instance data should be static

Code Smell

"Exception" should not be caught when not required by called methods

Code Smell

"sealed" classes should not have "protected" members

Code Smell

```
if (object.ReferenceEquals(this, other)) // base.Equals
      return true;
    return this.baseField == ((Base)other).baseField;
class Derived : Base
  private int derivedField;
  public override bool Equals(object other)
    if (object.ReferenceEquals(this, other))
      return true;
   }
    return base.Equals(other) && this.derivedField == ((Deri
}
```

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