

- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- C#**
- CSS
- Flex
- Go
- HTML
- Java
- JavaScript
- Kotlin
- Objective C
- PHP
- PL/I
- PL/SQL
- Python
- RPG
- Ruby
- Scala
- Swift
- Terraform
- Text
- TypeScript
- T-SQL
- VB.NET
- VB6
- XML



C# static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your C# code

All rules 409

Vulnerability 34

Bug 76

Security Hotspot 28

Code Smell 271

Quick Fix 52

Tags ▾

Search by name... 🔍

"protected" members

Code Smell

Underscores should be used to make large numbers readable

Code Smell

"ToString()" calls should not be redundant

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

"Generic.List" instances should not be part of public APIs

Analyze your code

Code Smell Major api-design

`System.Collections.Generic.List<T>` is a generic collection that is designed for performance and not inheritance. For example, it does not contain virtual members that make it easier to change the behavior of an inherited class. That means that future attempts to expand the behavior will be spoiled because the extension points simply aren't there. Instead, one of the following generic collections should be used:

- `System.Collections.Generic.IEnumerable<T>`
- `System.Collections.Generic.IReadOnlyCollection<T>`
- `System.Collections.Generic ICollection<TKey>`
- `System.Collections.Generic.IReadOnlyList<T>`
- `System.Collections.Generic.IList<TKey>`
- `System.Collections.ObjectModel.Collection<T>`
- `System.Collections.ObjectModel.ReadOnlyCollection<T>`
- `System.Collections.ObjectModel.KeyedCollection<TKey, Titem>`

This rule raises an issue every time a `System.Collections.Generic.List<T>` is exposed:

- As an externally visible member.
- As the return type of an externally visible method.
- As a parameter type of an externally visible method.

Noncompliant Code Example

```
namespace Foo
{
    public class Bar
    {
        public List<T> Method1(T arg) // Noncompliant
        {
            //...
        }
    }
}
```

Compliant Solution

```
namespace Foo
{
    public class Bar
    {
        public Collection<T> Method1(T arg)
        {
            //...
        }
    }
}
```

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

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](#)