sonar RULES

Products

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...

Strings should be normalized to uppercase
Code Smell

Exceptions should provide standard constructors
Code Smell

Assemblies should be marked with "NeutralResourcesLanguageAttribute"
Code Smell

Interfaces should not be empty
Code Smell

Enumerations should have "Int32" storage
Code Smell

Generic methods should provide type parameters
Code Smell

Multidimensional arrays should not be used
Code Smell

"static readonly" constants should be "const" instead
Code Smell

Strings or integral types should be used for indexers
Code Smell

Parameter names should not duplicate the names of their methods
Code Smell

Track use of "NotImplementedException"
Code Smell

Empty "default" clauses should be removed

Collapsible "if" statements should be merged

Analyze your code

Code Smell Major clumsy

Merging collapsible `if` statements increases the code’s readability.

Noncompliant Code Example

```
if (condition1)
{
    if (condition2)
    {
        // ...
    }
}
```

Compliant Solution

```
if (condition1 && condition2)
{
    // ...
}
```

Available In:

sonarlint | sonarcloud | sonarqube

Code Smell

Redundant property names should be omitted in anonymous classes

Code Smell

Declarations and initializations should be as concise as possible

Code Smell

Default parameter values should not be passed as arguments

Code Smell

Constructor and destructor declarations should not be redundant