sonar RULES Products

- 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

TypeScript static code analysis

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

All rules 279 | Vulnerability 27 | Bug 51 | Security Hotspot 43 | Code Smell 158 | Quick Fix 50

Tags | Search by name...

added
Code Smell

Primitive types should be omitted from initialized or defaulted declarations
Code Smell

Non-null assertions should not be used
Code Smell

"undefined" should not be assigned
Code Smell

Trailing commas should not be used
Code Smell

Array constructors should not be used
Code Smell

Quotes for string literals should be used consistently
Code Smell

Statements should end with semicolons
Code Smell

Comments should not be located at the end of lines of code
Code Smell

Loops should not contain more than a single "break" or "continue" statement
Code Smell

Variable, property and parameter names should comply with a naming convention
Code Smell

Lines should not end with trailing whitespaces
Code Smell

Union types should not have too many elements

Analyze your code

Code Smell | Major | brain-overload

Union types represent a value that can be one of the several types. When a union type is used for a function parameter and it is accepting too many types, it may indicate the function is having too many responsibilities. Sometimes it's worth creating a type alias for this union type. In all cases, the code should be reviewed and refactored to make it more maintainable.

Noncompliant Code Example

With the default threshold of 3:

```
let x: MyType1 | MyType2 | MyType3 | MyType4; // Noncomplian

function foo(p1: string, p2: MyType1 | MyType2 | MyType3 | M
    // ...
}
```

Compliant Solution

```
type MyUnionType = MyType1 | MyType2 | MyType3 | MyType4; //
let x: MyUnionType;

function foo(value: string, padding: MyUnionType) {
    // ...
}
```

Exceptions

This rule ignores union types part of `type` statement:

```
type MyUnionType = MyType1 | MyType2 | MyType3 | MyType4;
```

Available In:
sonarlint sonarcloud sonarqube

Files should contain an empty newline at the end

Code Smell

An open curly brace should be located at the end of a line

Code Smell

Tabulation characters should not be used

Code Smell

Function and method names should comply with a naming convention

Code Smell