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...

Code Smell

Variables should be used in the blocks where they are declared
Code Smell

Parameters should be passed in the correct order
Code Smell

Two branches in a conditional structure should not have exactly the same implementation
Code Smell

Unused assignments should be removed
Code Smell

Function parameters with default values should be last
Code Smell

Functions should not be defined inside loops
Code Smell

"switch" statements should not have too many "case" clauses
Code Smell

Only "while", "do", "for" and "switch" statements should be labelled
Code Smell

Sections of code should not be commented out
Code Smell

Unused function parameters should be removed
Code Smell

Track uses of "FIXME" tags
Code Smell

Constructors should not be declared inside interfaces

Analyze your code

Bug Major

Declare a constructor inside an `interface`, and you will get a simple method with the name "constructor". The same thing will happen if you create a `new` method inside the `interface`: you'll get a simple method named "new".

Instead, the intent was probably to specify that the type did not originate from a TypeScript file. In such cases, just use the `declare class` syntax.

Noncompliant Code Example

```
interface TypeDeclaredElsewhere {
  someMethod(): number;
  new(b: boolean): TypeDeclaredElsewhere; // Noncompliant
  constructor(b: boolean): void; // Noncompliant
}
```

Compliant Solution

```
declare class TypeDeclaredElsewhere {
  someMethod(): number;
  constructor(b: boolean);
}
```

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

Assignments should not be made from within sub-expressions

Code Smell

Labels should not be used

Code Smell

Variables should not be shadowed

Code Smell

Redundant pairs of parentheses should be removed

Code Smell