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...

reason to succeed
Code Smell

Single-character alternations in regular expressions should be replaced with character classes
Code Smell

Reluctant quantifiers in regular expressions should be followed by an expression that can't match the empty string
Code Smell

Tests should check which exception is thrown
Code Smell

Character classes in regular expressions should not contain the same character twice
Code Smell

Names of regular expressions named groups should be used
Code Smell

Regular expressions should not be too complicated
Code Smell

Optional property declarations should not use both '?' and 'undefined' syntax
Code Smell

Shorthand promises should be used
Code Smell

Template literals should not be nested
Code Smell

"undefined" should not be passed as the value of optional parameters
Code Smell

"in" should not be used on arrays

Functions should not be empty

Analyze your code

Code Smell Critical Quick Fix suspicious

There are several reasons for a function not to have a function body:

- It is an unintentional omission, and should be fixed to prevent an unexpected behavior in production.
- It is not yet, or never will be, supported. In this case an exception should be thrown in languages where that mechanism is available.
- The method is an intentionally-blank override. In this case a nested comment should explain the reason for the blank override.

Noncompliant Code Example

```
function foo() {
}

var foo = () => {};
```

Compliant Solution

```
function foo() {
    // This is intentional
}

var foo = () => {
    do_something();
};
```

Available In:

sonarlint sonarcloud sonarqube

Code Smell

Assignments should not be redundant

Code Smell

Functions should not have identical implementations

Code Smell

Sparse arrays should not be declared

Code Smell

Array-mutating methods should not be used misleadingly

Code Smell