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...

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

Collection and array contents should be used
Code Smell

Literals should not be thrown
Code Smell

Array indexes should be numeric
Code Smell

Assertion arguments should be passed in the correct order
Code Smell

Ternary operators should not be nested
Code Smell

"delete" should not be used on arrays
Code Smell

Variables and functions should not be redeclared
Code Smell

"indexOf" checks should not be for positive numbers
Code Smell

"arguments.caller" and "arguments.callee" should not be used
Code Smell

Multiline blocks should be enclosed in

Alternation in regular expressions should not contain empty alternatives

Analyze your code

Bug Major regex

Alternation is used to match a single regular expression out of several possible regular expressions. If one of the alternatives is empty it would match any input, which is most probably a mistake.

Noncompliant Code Example

```
/Jack|Peter|/.test('John'); // Noncompliant - returns 'true'
/Jack||Peter/.test('John'); // Noncompliant - returns 'true'
```

Compliant Solution

```
/Jack|Peter/.test('John'); // returns 'false'
```

Exceptions

One could use an empty alternation to make a regular expression group optional. Rule will not report on such cases.

```
/mandatory(-optional|)/.test('mandatory'); // returns 'true'
/mandatory(-optional|)/.test('mandatory-optional'); // retur
```

However, if there is a quantifier after the group the issue will be reported as using both `|` and quantifier is redundant.

```
/mandatory(-optional|)?/.test('mandatory'); // Noncompliant
```

Available In:

sonarlint sonarcloud sonarqube

curly braces

Code Smell

Boolean expressions should not be gratuitous

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