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

JavaScript static code analysis

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

All rules 285 | Vulnerability 29 | Bug 62 | Security Hotspot 43 | Code Smell 151 | Quick Fix 41

Tags

Search by name...

Code Smell

"if ... else if" constructs should end with "else" clauses

Code Smell

Control structures should use curly braces

Code Smell

String literals should not be duplicated

Code Smell

Expressions should not be too complex

Code Smell

Local storage should not be used

Vulnerability

Template literal placeholder syntax should not be used in regular strings

Bug

Built-in objects should not be overridden

Bug

"for...in" loops should filter properties before acting on them

Bug

Results of operations on strings should not be ignored

Bug

Increment (++) and decrement (--) operators should not be used in a method call or mixed with other operators in an expression

Code Smell

"for in" should not be used with iterables

Code Smell

Functions should not have identical implementations

Analyze your code

Code Smell Major confusing duplicate suspicious

When two functions have the same implementation, either it was a mistake - something else was intended - or the duplication was intentional, but may be confusing to maintainers. In the latter case, the code should be refactored.

Noncompliant Code Example

```
function calculateCode() {
  doTheThing();
  doOtherThing();
  return code;
}

function getName() {  // Noncompliant
  doTheThing();
  doOtherThing();
  return code;
}
```

Compliant Solution

```
function calculateCode() {
  doTheThing();
  doOtherThing();
  return code;
}

function getName() {
  return calculateCode();
}
```

Exceptions

Functions with fewer than 3 lines are ignored.

Available In:

sonarlint sonarcloud sonarqube

Functions should use "return" consistently

Code Smell

Variables and functions should not be declared in the global scope

Code Smell

Arithmetic operators should only have numbers as operands

Code Smell

Values not convertible to numbers should not be used in numeric comparisons

Code Smell