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...

Object literal shorthand syntax should be used
Code Smell

Strings and non-strings should not be added
Code Smell

Object literal syntax should 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

Variables should be defined before being used

Analyze your code

Bug Blocker

When a non-existent variable is referenced a `ReferenceError` is raised.

Due to the dynamic nature of JavaScript this can happen in a number of scenarios:

- When typo was made in a symbol's name.
- When using variable declared with `let` or `const` before declaration (unlike `var`-declarations, they are not hoisted to the top of the scope).
- Due to confusion with scopes of `let`- and `const`-declarations (they have block scope, unlike `var`-declarations, having function scope).
- When accessing a property in the wrong scope (e.g. forgetting to specify `this.`).

This rule does not raise issues on global variables which are defined with `sonar.javascript.globals` and `sonar.javascript.environments` properties.

Noncompliant Code Example

```
var john = {
  firstName: "john",
  show: function() { console.log(firstName); } // Noncomplia
}
john.show();
```

Compliant Solution

```
var john = {
  firstName: "john",
  show: function() { console.log(this.firstName); }
}
john.show();
```

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