JavaScript static code analysis

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

Code Smell

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

Multiline blocks should be enclosed in 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

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

Errors should not be created without being thrown

Analyze your code

Bug Major Quick Fix error-handling

Creating a new `Error` without actually throwing it is useless and is probably due to a mistake.

Noncompliant Code Example

```
if (x < 0) {
  new Error("x must be nonnegative");
}
```

Compliant Solution

```
if (x < 0) {
  throw new Error("x must be nonnegative");
}
```

Available In:
sonarlint sonarcloud sonarqube

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

Assignments should not be made from within sub-expressions