JavaScript static code analysis

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

Regular expressions should not contain empty groups
Code Smell

Regular expressions should not contain multiple spaces
Code Smell

Chai assertions should have only one 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

Shorthand promises should be used
Code Smell

Template literals should not be nested
Code Smell

"for" loop increment clauses should modify the loops' counters

Analyze your code

Code Smell Critical confusing

It can be extremely confusing when a `for` loop's counter is incremented outside of its increment clause. In such cases, the increment should be moved to the loop's increment clause if at all possible.

Noncompliant Code Example

```
for (i = 0; i < 10; j++) { // Noncompliant
  // ...
  i++;
}
```

Compliant Solution

```
for (i = 0; i < 10; i++, j++) {
  // ...
}
```

Or

```
for (i = 0; i < 10; i++) {
  // ...
  j++;
}
```

Available In:

sonarlint sonarcloud sonarqube

"in" should not be used on arrays

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