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...

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
Code Smell

Collection and array contents should be used
Code Smell

Functions should always return the same type
Code Smell

Arguments to built-in functions should match documented types
Code Smell

Literals should not be thrown
Code Smell

Functions should not be called both with and without "new"
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

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

Privacy Policy

"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