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

TS TypeScript static code analysis

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

All rules 279	Vulnerability 27	Bug 51	Security Hotspot 43	Code Smell 158	Quick Fix 50

Tags ⌄ | Search by name... 🔍

---

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
⊘ 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
⊘ Code Smell

Labels should not be used
⊘ Code Smell

Variables should not be shadowed
⊘ Code Smell

Redundant pairs of parentheses should be removed
⊘ Code Smell

Nested blocks of code should not be

All branches in a conditional structure should not have exactly the same implementation

Analyze your code

🐛 Bug   🔺 Major ⓘ

Having all branches in a `switch` or `if` chain with the same implementation is an error. Either a copy-paste error was made and something different should be executed, or there shouldn't be a `switch`/`if` chain at all.

Noncompliant Code Example

```
if (b == 0) {  // Noncompliant
  doOneMoreThing();
}
else {
  doOneMoreThing();
}

let a = (b == 0) ? getValue() : getValue();    // Noncompliant

switch (i) {  // Noncompliant
  case 1:
    doSomething();
    break;
  case 2:
    doSomething();
    break;
  case 3:
    doSomething();
    break;
  default:
    doSomething();
}
```

Exceptions

This rule does not apply to `if` chains without `else`-s, or to `switch`-es without `default` clauses.

```
if(b == 0) {    //no issue, this could have been done on pur
  doSomething();
} else if(b == 1) {
  doSomething();
}
```

Available In:

sonarlint 😀 | sonarcloud ☁ | sonarqube

5/29/22, 5:35 PM

TypeScript static code analysis: All branches in a conditional structure should not have exactly the same implementation

Nested blocks of code should not be
left empty

⚠ Code Smell

Functions should not have too many
parameters

⚠ Code Smell

OS commands should not be
vulnerable to argument injection
attacks

🔒 Vulnerability

Repeated patterns in regular
expressions should not match the
empty string

🐛 Bug