




 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 rules285

Vulnerability29

Bug62

Security Hotspot43

Code Smell151

Quick Fix41

Tags ▾

Search by name... 🔍

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

Code Smell

Sections of code should not be commented out

Code Smell

Unused function parameters should be removed

Code Smell

### Destructuring patterns should not be empty

Analyze your code

BugMajor?

Destructuring is a convenient way of extracting multiple values from data stored in (possibly nested) objects and arrays. However, it is possible to create an empty pattern that has no effect. When empty curly braces or brackets are used to the right of a property name most of the time the intent was to use a default value instead.

This rule raises an issue when empty destructuring pattern is used.

#### Noncompliant Code Example

```
var {a: {}, b} = myObj; // Noncompliant
function foo({first: [], second}) { // Noncompliant
  // ...
}
```

#### Compliant Solution

```
var {a = {}, b} = myObj;
function foo({first = [], second}) {
  // ...
}
```

Available In:

sonarlint | sonarcloud | sonarqube

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.  
[Privacy Policy](#)

https://rules.sonarsource.com/javascript/RSPEC-3799

1/2

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