

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

Code Smell
A new session should be created during user authentication
Vulnerability
JWT should be signed and verified with strong cipher algorithms
Vulnerability
Cipher algorithms should be robust
Vulnerability
Encryption algorithms should be used with secure mode and padding scheme
Vulnerability
Server hostnames should be verified during SSL/TLS connections
Vulnerability
Server certificates should be verified during SSL/TLS connections
Vulnerability
Cryptographic keys should be robust
Vulnerability
Weak SSL/TLS protocols should not be used
Vulnerability
Origins should be verified during cross-origin communications
Vulnerability
Regular expressions should not be vulnerable to Denial of Service attacks
Vulnerability
File uploads should be restricted
Vulnerability
Function calls should not pass extra

Tags ▾

Search by name... 🔍

Callbacks of array methods should have return statements

Analyze your code

Bug

Blocker

Arrays in JavaScript have several methods for filtering, mapping, or folding that require a callback. Not having a return statement in such a callback function is most likely a mistake, because processing of the array uses the return value of the callback. If there is no return, callback will implicitly return `undefined`, which will likely fail.

This rule applies for the following methods of an array:

- Array.from
- Array.prototype.every
- Array.prototype.filter
- Array.prototype.find
- Array.prototype.findIndex
- Array.prototype.map
- Array.prototype.reduce
- Array.prototype.reduceRight
- Array.prototype.some
- Array.prototype.sort

Noncompliant Code Example





```
let arr = ["a", "b", "c"];
let merged = arr.reduce(function(a, b) {
  a.concat(b);
}); // Noncompliant: No return statement, will result in Typ
```

Compliant Solution

```
let arr = ["a", "b", "c"];
let merged = arr.reduce(function(a, b) {
  return a.concat(b);
}); // merged === "abc"
```

Available In:

sonarlint | sonarcloud | sonarqube

arguments  Bug
Regular expressions should be syntactically valid  Bug
Getters and setters should access the expected fields  Bug
"super()" should be invoked appropriately  Bug
"Symbol" should not be used as a