




 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



## 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... 🔍

Common Closure Statement (CCS) braces

Code Smell

String literals should not be duplicated

Code Smell

Expressions should not be too complex

Code Smell

Template literal placeholder syntax should not be used in regular strings

Bug

Built-in objects should not be overridden

Bug

"for...in" loops should filter properties before acting on them

Bug

Results of operations on strings should not be ignored

Bug

Increment (++) and decrement (--) operators should not be used in a method call or mixed with other operators in an expression

Code Smell

Optional boolean parameters should have default value

Code Smell

Union types should not have too many elements

Code Smell

Dependencies should be explicit

Code Smell

"this" should not be assigned to variables

Code Smell

Sparse arrays should not be declared

Analyze your code

Code Smell

Major

suspicious

An array declared with missing ("sparse") elements is likely to be an error: an extra comma was inserted or perhaps the developer meant to insert the missing value and forgot.

Noncompliant Code Example

```
let a = [1, , 3, 6, 9]; // Noncompliant
```

Compliant Solution

```
let a = [1, 3, 6, 9];
```

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/typescript/RSPEC-4140

1/2

 Code Smell
<b>The "any" type should not be used</b>  Code Smell
<b>"for in" should not be used with iterables</b>  Code Smell
<b>Functions should use "return" consistently</b>  Code Smell
<b>"arguments" should not be accessed directly</b>  Code Smell