

































-  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 rules279

Vulnerability27

Bug51

Security Hotspot43

Code Smell158

Quick Fix50

Tags ▾

Search by name... 🔍

Function declarations should not be made within blocks

Bug

Writing cookies is security-sensitive

Security Hotspot

"continue" should not be used

Code Smell

Primitive return types should be used

Code Smell

Default type parameters should be omitted

Code Smell

Type assertions should use "as"

Code Smell

Method overloads should be grouped together

Code Smell

Interfaces should not be empty

Code Smell

Trailing commas should be used

Code Smell

"import" should be used to include external code

Code Smell

Braces and parentheses should be used consistently with arrow functions

Code Smell

Destructuring syntax should be used for assignments

Code Smell

Template strings should be used instead of concatenation

"switch" statements should not have too many "case" clauses

Analyze your code

Code Smell

Major ?

brain-overload

When switch statements have large sets of case clauses, it is usually an attempt to map two sets of data. A real map structure would be more readable and maintainable, and should be used instead.

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](#)

 Code Smell
Shorthand object properties should be grouped at the beginning or end of an object declaration  Code Smell
Object literal shorthand syntax should be used  Code Smell
Strings and non-strings should not be added  Code Smell
Primitive types should be omitted