

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

**TS** **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 rules279

Vulnerability27

Bug51

Security Hotspot43

Code Smell158

Quick Fix50

Tags ▾

Search by name... 🔍

Types without members, 'any' and 'never' should not be used in type intersections

Bug

Getters and setters should access the expected fields

Bug

"super()" should be invoked appropriately

Bug

Results of "in" and "instanceof" should be negated rather than operands

Bug

A compare function should be provided when using "Array.prototype.sort()"

Bug

Jump statements should not occur in "finally" blocks

Bug

Using slow regular expressions is security-sensitive

Security Hotspot

Using publicly writable directories is security-sensitive

Security Hotspot

Using clear-text protocols is security-sensitive

Security Hotspot

Expanding archive files without controlling resource consumption is security-sensitive

Security Hotspot

Using weak hashing algorithms is security-sensitive

Security Hotspot

Tests should include assertions

Analyze your code

Code Smell

Blocker

tests chai mocha

A test case without assertions ensures only that no exceptions are thrown. Beyond basic runnability, it ensures nothing about the behavior of the code under test.

This rule raises an exception when the assertion library chai is imported but no assertion is used in a test.

Noncompliant Code Example

```
const expect = require('chai').expect;

describe("No assertion", function() {
  it("doesn't test anything", function() { // Noncompliant
    const str = "";
  });
});
```

Compliant Solution

```
const expect = require('chai').expect;

describe("Has assertions", function() {
  it("tests a string", function() {
    const str = "";
    expect(str).to.be.a('string');
  });
});
```

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-2699

1/2

<div>Disabling CSRF protections is security-sensitive</div> <div> Security Hotspot</div>
<div>Using pseudorandom number generators (PRNGs) is security-sensitive</div> <div> Security Hotspot</div>
<div>Dynamically executing code is security-sensitive</div> <div> Security Hotspot</div>
<div>Equality operators should not be used in "for" loop termination conditions</div>