Products >

O Quick 50 Fix

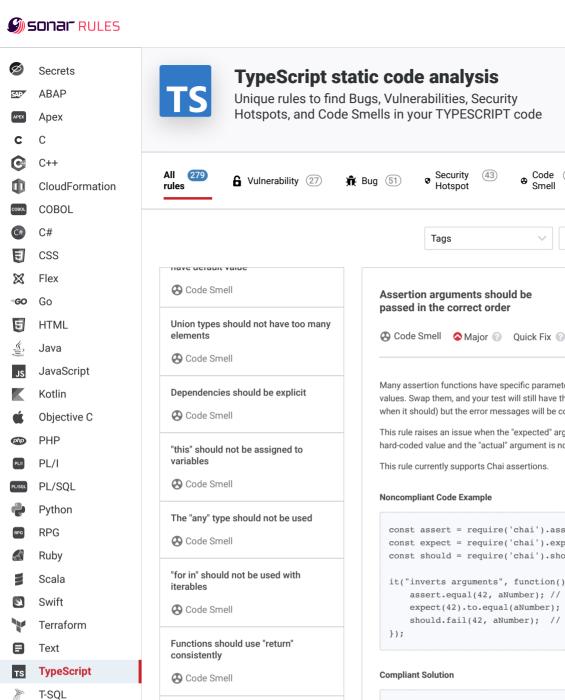
Analyze your code

tests chai suspicious

Q

(158)

Search by name.



"arguments" should not be accessed

Comparison operators should not be

Private properties that are only

assigned in the constructor or at

declaration should be "readonly"

Property getters and setters should

directly

Code Smell

used with strings

Code Smell

Code Smell

come in pairs

Code Smell

Code Smell

JavaScript parser failure

Many assertion functions have specific parameters for the expected and actual values. Swap them, and your test will still have the same outcome (succeed/fail when it should) but the error messages will be confusing. This rule raises an issue when the "expected" argument of an assertion function is a hard-coded value and the "actual" argument is not. This rule currently supports Chai assertions. const assert = require('chai').assert; const expect = require('chai').expect; const should = require('chai').should(); it("inverts arguments", function() { assert.equal(42, aNumber); // Noncompliant expect(42).to.equal(aNumber); // Noncompliant should.fail(42, aNumber); // Noncompliant const assert = require('chai').assert; const expect = require('chai').expect; const should = require('chai').should(); it("inverts arguments", function() { assert.equal(aNumber, 42): expect(aNumber).to.equal(42); should.fail(aNumber, 42); }); 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.

VB.NET

VB6

XML

The ternary operator should not be used

Code Smell

"===" and "!==" should be used instead of "==" and "!="

Code Smell

Functions should not have too many lines of code

Code Smell

Track comments matching a regular expression

Code Smell