TypeScript static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your TYPESCRIPT code

the same argument
Bug

Alternatives in regular expressions should be grouped when used with anchors
Bug

Promise rejections should not be caught by 'try' block
Bug

Collection elements should not be replaced unconditionally
Bug

Constructors should not be declared inside interfaces
Bug

Errors should not be created without being thrown
Bug

Collection sizes and array length comparisons should make sense
Bug

All branches in a conditional structure should not have exactly the same implementation
Bug

Destructuring patterns should not be empty
Bug

The output of functions that don't return anything should not be used
Bug

Comma and logical OR operators should not be used in switch cases
Bug

Generators should "yield" something

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

Analyze your code

Bug Critical Quick Fix bad-practice

The default sort order is alphabetic, rather than numeric, regardless of the types in the array. Specifically, even if an array contains only numbers, all values in it will be converted to strings and sorted lexicographically, for an order like this: 1, 15, 2, 20, 5.

Fortunately the `sort` method allows you to pass an optional compare function to specify the sort order. When a compare function is supplied, the returned order depends on the return value of the compare function.

Noncompliant Code Example

```
var myarray = [80, 3, 9, 34, 23, 5, 1];

myarray.sort();
console.log(myarray); // outputs: [1, 23, 3, 34, 5, 80, 9]
```

Compliant Solution

```
var myarray = [80, 3, 9, 34, 23, 5, 1];

myarray.sort((a, b) => (a - b));
console.log(myarray); // outputs: [1, 3,  5, 9, 23, 34, 80]
```

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

Bug

"new" operators should be used with functions

Bug

Non-existent operators '=+', '=-' and '=!' should not be used

Bug

"NaN" should not be used in comparisons

Bug

A "for" loop update clause should move the counter in the right direction