TypeScript static code analysis

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

lines
Code Smell

Magic numbers should not be used
Code Smell

Collapsible "if" statements should be merged
Code Smell

Standard outputs should not be used directly to log anything
Code Smell

Files should not have too many lines of code
Code Smell

Lines should not be too long
Code Smell

Debugger statements should not be used
Vulnerability

Regular expressions using Unicode character classes or property escapes should enable the unicode flag
Bug

The base should be provided to "parseInt"
Bug

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

Parameters should be passed in the correct order

Analyze your code

Code Smell Major

When the names of arguments in a function call match the names of the function parameters, it contributes to clearer, more readable code. However, when the names match, but are passed in a different order than the function parameters, it indicates a mistake in the parameter order which will likely lead to unexpected results.

Noncompliant Code Example

```
function divide(divisor, dividend) {
  return divisor/dividend;
}

function doTheThing() {
  var divisor = 15;
  var dividend = 5;

  var result = divide(dividend, divisor);  // Noncompliant;
  //...
}
```

Compliant Solution

```
function divide(divisor, dividend) {
  return divisor/dividend;
}

function doTheThing() {
  var divisor = 15;
  var dividend = 5;

  var result = divide(divisor, dividend);
  //...
}
```

Exceptions

Swapped arguments that are compared beforehand in an enclosing `if`-statement are ignored:

```
function divide(divisor, dividend) {
  return divisor/dividend;
}

function doTheThing() {
  var divisor = 15;
  var dividend = 5;
  if (divisor > dividend) {
    var result = divide(dividend, divisor);
    //...
```

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

```
    }
  }
```

Available In:

sonarlint | sonarcloud | sonarqube