




 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... 🔍


 Code Smell


Primitive types should be omitted from initialized or defaulted declarations

 Code Smell


 Code Smell


Non-null assertions should not be used

 Code Smell


 Code Smell


"undefined" should not be assigned

 Code Smell


 Code Smell


Trailing commas should not be used

 Code Smell


 Code Smell

Array constructors should not be used

 Code Smell

 Code Smell

Quotes for string literals should be used consistently

 Code Smell

 Code Smell

Statements should end with semicolons

 Code Smell

 Code Smell

Comments should not be located at the end of lines of code

 Code Smell

 Code Smell

Loops should not contain more than a single "break" or "continue" statement

 Code Smell

 Code Smell

Variable, property and parameter names should comply with a naming convention

 Code Smell


 Code Smell


Lines should not end with trailing whitespaces


 Code Smell

Interfaces should not be empty

Analyze your code

 Code Smell

 Minor ?

 Quick Fix ?

An empty interface is equivalent to an empty object ('{}'). Normally you cannot directly assign an object literal to a type when the object literal contains more properties than are specified in the type. But in the case of an empty interface, this check is not done, and such assignments will be successful. The result is highly likely to confuse maintainers.




Noncompliant Code Example

```
interface MyFace {} // Noncompliant
```

Compliant Solution

```
interface MyFace {
  foo: number;
}
```

Available In:

 |  | 

© 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-4023

1/2

<div>Files should contain an empty newline at the end</div> <div> Code Smell</div>
<div>An open curly brace should be located at the end of a line</div> <div> Code Smell</div>
<div>Tabulation characters should not be used</div> <div> Code Smell</div>
<div>Function and method names should comply with a naming convention</div> <div> Code Smell</div>