## TypeScript static code analysis

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

[HTTP responses should not be vulnerable to session fixation](https://rules.sonarsource.com/typescript/RSPEC-6287)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-6287)

[DOM updates should not lead to open redirect vulnerabilities](https://rules.sonarsource.com/typescript/RSPEC-6105)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-6105)

[Extracting archives should not lead to zip slip vulnerabilities](https://rules.sonarsource.com/typescript/RSPEC-6096)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-6096)

[DOM updates should not lead to cross-site scripting (XSS) attacks](https://rules.sonarsource.com/typescript/RSPEC-5696)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-5696)

[Dynamic code execution should not be vulnerable to injection attacks](https://rules.sonarsource.com/typescript/RSPEC-5334)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-5334)

[NoSQL operations should not be vulnerable to injection attacks](https://rules.sonarsource.com/typescript/RSPEC-5147)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-5147)

[HTTP request redirections should not be open to forging attacks](https://rules.sonarsource.com/typescript/RSPEC-5146)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-5146)

[Endpoints should not be vulnerable to reflected cross-site scripting (XSS) attacks](https://rules.sonarsource.com/typescript/RSPEC-5131)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-5131)

[Database queries should not be vulnerable to injection attacks](https://rules.sonarsource.com/typescript/RSPEC-3649)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-3649)

[XML parsers should not be vulnerable to XXE attacks](https://rules.sonarsource.com/typescript/RSPEC-2755)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-2755)

[I/O function calls should not be vulnerable to path injection attacks](https://rules.sonarsource.com/typescript/RSPEC-2083)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-2083)

[OS commands should not be vulnerable to command injection attacks](https://rules.sonarsource.com/typescript/RSPEC-2076)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-2076)

[Disabling Vue.js built-in escaping is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-6299)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-6299)

[Disabling Angular built-in sanitization is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-6268)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-6268)

[Hard-coded credentials are security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-2068)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-2068)

[Function returns should not be invariant](https://rules.sonarsource.com/typescript/RSPEC-3516)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3516)

[Assertions should be complete](https://rules.sonarsource.com/typescript/RSPEC-2970)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-2970)

[Tests should include assertions](https://rules.sonarsource.com/typescript/RSPEC-2699)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-2699)

[Octal values should not be used](https://rules.sonarsource.com/typescript/RSPEC-1314)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1314)

[Switch cases should end with an unconditional "break" statement](https://rules.sonarsource.com/typescript/RSPEC-128)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-128)

["switch" statements should not contain non-case labels](https://rules.sonarsource.com/typescript/RSPEC-1219)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1219)

[A new session should be created during user authentication](https://rules.sonarsource.com/typescript/RSPEC-5876)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-5876)

[JWT should be signed and verified with strong cipher algorithms](https://rules.sonarsource.com/typescript/RSPEC-5659)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-5659)

[Cipher algorithms should be robust](https://rules.sonarsource.com/typescript/RSPEC-5547)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-5547)

[Encryption algorithms should be used with secure mode and padding scheme](https://rules.sonarsource.com/typescript/RSPEC-5542)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-5542)

[Server hostnames should be verified during SSL/TLS connections](https://rules.sonarsource.com/typescript/RSPEC-5527)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-5527)

[Server certificates should be verified during SSL/TLS connections](https://rules.sonarsource.com/typescript/RSPEC-4830)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-4830)

[Cryptographic keys should be robust](https://rules.sonarsource.com/typescript/RSPEC-4426)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-4426)

[Weak SSL/TLS protocols should not be used](https://rules.sonarsource.com/typescript/RSPEC-4423)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-4423)

[Origins should be verified during cross-origin communications](https://rules.sonarsource.com/typescript/RSPEC-2819)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-2819)

[Regular expressions should not be vulnerable to Denial of Service attacks](https://rules.sonarsource.com/typescript/RSPEC-2631)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-2631)

[File uploads should be restricted](https://rules.sonarsource.com/typescript/RSPEC-2598)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-2598)

[Regular expressions should be syntactically valid](https://rules.sonarsource.com/typescript/RSPEC-5856)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-5856)

[Types without members, 'any' and 'never' should not be used in type intersections](https://rules.sonarsource.com/typescript/RSPEC-4335)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-4335)

[Getters and setters should access the expected fields](https://rules.sonarsource.com/typescript/RSPEC-4275)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-4275)

["super()" should be invoked appropriately](https://rules.sonarsource.com/typescript/RSPEC-3854)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-3854)

[Results of "in" and "instanceof" should be negated rather than operands](https://rules.sonarsource.com/typescript/RSPEC-3812)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-3812)

[A compare function should be provided when using "Array.prototype.sort()"](https://rules.sonarsource.com/typescript/RSPEC-2871)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-2871)

[Jump statements should not occur in "finally" blocks](https://rules.sonarsource.com/typescript/RSPEC-1143)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-1143)

[Using slow regular expressions is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5852)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5852)

[Using publicly writable directories is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5443)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5443)

[Using clear-text protocols is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5332)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5332)

[Expanding archive files without controlling resource consumption is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5042)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5042)

[Using weak hashing algorithms is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-4790)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-4790)

[Disabling CSRF protections is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-4502)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-4502)

[Using pseudorandom number generators (PRNGs) is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-2245)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-2245)

[Dynamically executing code is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-1523)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-1523)

[Equality operators should not be used in "for" loop termination conditions](https://rules.sonarsource.com/typescript/RSPEC-888)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-888)

[Tests should not execute any code after "done()" is called](https://rules.sonarsource.com/typescript/RSPEC-6079)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-6079)

[Union and intersection types should not be defined with duplicated elements](https://rules.sonarsource.com/typescript/RSPEC-4621)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4621)

["default" clauses should be last](https://rules.sonarsource.com/typescript/RSPEC-4524)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4524)

["await" should only be used with promises](https://rules.sonarsource.com/typescript/RSPEC-4123)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4123)

[A conditionally executed single line should be denoted by indentation](https://rules.sonarsource.com/typescript/RSPEC-3973)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3973)

[Conditionals should start on new lines](https://rules.sonarsource.com/typescript/RSPEC-3972)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3972)

[Cognitive Complexity of functions should not be too high](https://rules.sonarsource.com/typescript/RSPEC-3776)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3776)

["void" should not be used](https://rules.sonarsource.com/typescript/RSPEC-3735)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3735)

[Loop counters should not be assigned to from within the loop body](https://rules.sonarsource.com/typescript/RSPEC-2310)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-2310)

["for" loop increment clauses should modify the loops' counters](https://rules.sonarsource.com/typescript/RSPEC-1994)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1994)

[Functions should not be empty](https://rules.sonarsource.com/typescript/RSPEC-1186)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1186)

[Server-side requests should not be vulnerable to forging attacks](https://rules.sonarsource.com/typescript/RSPEC-5144)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-5144)

[Non-empty statements should change control flow or have at least one side-effect](https://rules.sonarsource.com/typescript/RSPEC-905)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-905)

[Regular expressions with the global flag should be used with caution](https://rules.sonarsource.com/typescript/RSPEC-6351)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-6351)

[Replacement strings should reference existing regular expression groups](https://rules.sonarsource.com/typescript/RSPEC-6328)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-6328)

[Regular expressions should not contain control characters](https://rules.sonarsource.com/typescript/RSPEC-6324)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-6324)

[Alternation in regular expressions should not contain empty alternatives](https://rules.sonarsource.com/typescript/RSPEC-6323)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-6323)

[Mocha timeout should be disabled by setting it to "0".](https://rules.sonarsource.com/typescript/RSPEC-6080)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-6080)

[Unicode Grapheme Clusters should be avoided inside regex character classes](https://rules.sonarsource.com/typescript/RSPEC-5868)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-5868)

[Assertions should not be given twice the same argument](https://rules.sonarsource.com/typescript/RSPEC-5863)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-5863)

[Alternatives in regular expressions should be grouped when used with anchors](https://rules.sonarsource.com/typescript/RSPEC-5850)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-5850)

[Promise rejections should not be caught by 'try' block](https://rules.sonarsource.com/typescript/RSPEC-4822)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-4822)

[Collection elements should not be replaced unconditionally](https://rules.sonarsource.com/typescript/RSPEC-4143)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-4143)

[Constructors should not be declared inside interfaces](https://rules.sonarsource.com/typescript/RSPEC-4124)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-4124)

[Errors should not be created without being thrown](https://rules.sonarsource.com/typescript/RSPEC-3984)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-3984)

[Collection sizes and array length comparisons should make sense](https://rules.sonarsource.com/typescript/RSPEC-3981)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-3981)

[All branches in a conditional structure should not have exactly the same implementation](https://rules.sonarsource.com/typescript/RSPEC-3923)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-3923)

[Destructuring patterns should not be empty](https://rules.sonarsource.com/typescript/RSPEC-3799)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-3799)

[The output of functions that don't return anything should not be used](https://rules.sonarsource.com/typescript/RSPEC-3699)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-3699)

[Comma and logical OR operators should not be used in switch cases](https://rules.sonarsource.com/typescript/RSPEC-3616)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-3616)

[Generators should "yield" something](https://rules.sonarsource.com/typescript/RSPEC-3531)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-3531)

["new" operators should be used with functions](https://rules.sonarsource.com/typescript/RSPEC-2999)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-2999)

[Non-existent operators '=+', '=-' and '=!' should not be used](https://rules.sonarsource.com/typescript/RSPEC-2757)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-2757)

["NaN" should not be used in comparisons](https://rules.sonarsource.com/typescript/RSPEC-2688)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-2688)

[A "for" loop update clause should move the counter in the right direction](https://rules.sonarsource.com/typescript/RSPEC-2251)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-2251)

[Return values from functions without side effects should not be ignored](https://rules.sonarsource.com/typescript/RSPEC-2201)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-2201)

[Special identifiers should not be bound or assigned](https://rules.sonarsource.com/typescript/RSPEC-2137)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-2137)

[Values should not be uselessly incremented](https://rules.sonarsource.com/typescript/RSPEC-2123)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-2123)

[Related "if/else if" statements should not have the same condition](https://rules.sonarsource.com/typescript/RSPEC-1862)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-1862)

[Objects should not be created to be dropped immediately without being used](https://rules.sonarsource.com/typescript/RSPEC-1848)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-1848)

[Identical expressions should not be used on both sides of a binary operator](https://rules.sonarsource.com/typescript/RSPEC-1764)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-1764)

[All code should be reachable](https://rules.sonarsource.com/typescript/RSPEC-1763)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-1763)

[Loops with at most one iteration should be refactored](https://rules.sonarsource.com/typescript/RSPEC-1751)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-1751)

[Variables should not be self-assigned](https://rules.sonarsource.com/typescript/RSPEC-1656)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-1656)

[Bitwise operators should not be used in boolean contexts](https://rules.sonarsource.com/typescript/RSPEC-1529)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-1529)

[Constructing arguments of system commands from user input is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-6350)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-6350)

[Allowing requests with excessive content length is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5693)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5693)

[Statically serving hidden files is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5691)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5691)

[Using intrusive permissions is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5604)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5604)

[Disabling auto-escaping in template engines is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5247)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5247)

[Using shell interpreter when executing OS commands is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-4721)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-4721)

[Setting loose POSIX file permissions is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-2612)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-2612)

[Formatting SQL queries is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-2077)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-2077)

[Comma operator should not be used](https://rules.sonarsource.com/typescript/RSPEC-878)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-878)

[Regular expressions should not contain empty groups](https://rules.sonarsource.com/typescript/RSPEC-6331)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-6331)

[Regular expressions should not contain multiple spaces](https://rules.sonarsource.com/typescript/RSPEC-6326)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-6326)

[Chai assertions should have only one reason to succeed](https://rules.sonarsource.com/typescript/RSPEC-6092)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-6092)

[Single-character alternations in regular expressions should be replaced with character classes](https://rules.sonarsource.com/typescript/RSPEC-6035)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-6035)

[Reluctant quantifiers in regular expressions should be followed by an expression that can't match the empty string](https://rules.sonarsource.com/typescript/RSPEC-6019)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-6019)

[Tests should check which exception is thrown](https://rules.sonarsource.com/typescript/RSPEC-5958)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-5958)

[Character classes in regular expressions should not contain the same character twice](https://rules.sonarsource.com/typescript/RSPEC-5869)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-5869)

[Names of regular expressions named groups should be used](https://rules.sonarsource.com/typescript/RSPEC-5860)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-5860)

[Regular expressions should not be too complicated](https://rules.sonarsource.com/typescript/RSPEC-5843)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-5843)

[Optional property declarations should not use both '?' and 'undefined' syntax](https://rules.sonarsource.com/typescript/RSPEC-4782)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4782)

[Shorthand promises should be used](https://rules.sonarsource.com/typescript/RSPEC-4634)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4634)

[Template literals should not be nested](https://rules.sonarsource.com/typescript/RSPEC-4624)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4624)

["undefined" should not be passed as the value of optional parameters](https://rules.sonarsource.com/typescript/RSPEC-4623)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4623)

["in" should not be used on arrays](https://rules.sonarsource.com/typescript/RSPEC-4619)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4619)

[Assignments should not be redundant](https://rules.sonarsource.com/typescript/RSPEC-4165)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4165)

[Functions should not have identical implementations](https://rules.sonarsource.com/typescript/RSPEC-4144)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4144)

[Sparse arrays should not be declared](https://rules.sonarsource.com/typescript/RSPEC-4140)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4140)

[Array-mutating methods should not be used misleadingly](https://rules.sonarsource.com/typescript/RSPEC-4043)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4043)

[Collection and array contents should be used](https://rules.sonarsource.com/typescript/RSPEC-4030)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4030)

[Literals should not be thrown](https://rules.sonarsource.com/typescript/RSPEC-3696)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3696)

[Array indexes should be numeric](https://rules.sonarsource.com/typescript/RSPEC-3579)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3579)

[Assertion arguments should be passed in the correct order](https://rules.sonarsource.com/typescript/RSPEC-3415)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3415)

[Ternary operators should not be nested](https://rules.sonarsource.com/typescript/RSPEC-3358)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3358)

["delete" should not be used on arrays](https://rules.sonarsource.com/typescript/RSPEC-2870)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-2870)

[Variables and functions should not be redeclared](https://rules.sonarsource.com/typescript/RSPEC-2814)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-2814)

["indexOf" checks should not be for positive numbers](https://rules.sonarsource.com/typescript/RSPEC-2692)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-2692)

["arguments.caller" and "arguments.callee" should not be used](https://rules.sonarsource.com/typescript/RSPEC-2685)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-2685)

[Multiline blocks should be enclosed in curly braces](https://rules.sonarsource.com/typescript/RSPEC-2681)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-2681)

[Boolean expressions should not be gratuitous](https://rules.sonarsource.com/typescript/RSPEC-2589)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-2589)

[Variables should be used in the blocks where they are declared](https://rules.sonarsource.com/typescript/RSPEC-2392)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-2392)

[Parameters should be passed in the correct order](https://rules.sonarsource.com/typescript/RSPEC-2234)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-2234)

[Two branches in a conditional structure should not have exactly the same implementation](https://rules.sonarsource.com/typescript/RSPEC-1871)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1871)

[Unused assignments should be removed](https://rules.sonarsource.com/typescript/RSPEC-1854)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1854)

[Function parameters with default values should be last](https://rules.sonarsource.com/typescript/RSPEC-1788)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1788)

[Functions should not be defined inside loops](https://rules.sonarsource.com/typescript/RSPEC-1515)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1515)

["switch" statements should not have too many "case" clauses](https://rules.sonarsource.com/typescript/RSPEC-1479)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1479)

[Only "while", "do", "for" and "switch" statements should be labelled](https://rules.sonarsource.com/typescript/RSPEC-1439)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1439)

[Sections of code should not be commented out](https://rules.sonarsource.com/typescript/RSPEC-125)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-125)

[Unused function parameters should be removed](https://rules.sonarsource.com/typescript/RSPEC-1172)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1172)

[Track uses of "FIXME" tags](https://rules.sonarsource.com/typescript/RSPEC-1134)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1134)

[Assignments should not be made from within sub-expressions](https://rules.sonarsource.com/typescript/RSPEC-1121)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1121)

[Labels should not be used](https://rules.sonarsource.com/typescript/RSPEC-1119)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1119)

[Variables should not be shadowed](https://rules.sonarsource.com/typescript/RSPEC-1117)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1117)

[Redundant pairs of parentheses should be removed](https://rules.sonarsource.com/typescript/RSPEC-1110)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1110)

[Nested blocks of code should not be left empty](https://rules.sonarsource.com/typescript/RSPEC-108)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-108)

[Functions should not have too many parameters](https://rules.sonarsource.com/typescript/RSPEC-107)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-107)

[OS commands should not be vulnerable to argument injection attacks](https://rules.sonarsource.com/typescript/RSPEC-5883)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-5883)

[Repeated patterns in regular expressions should not match the empty string](https://rules.sonarsource.com/typescript/RSPEC-5842)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-5842)

[Empty collections should not be accessed or iterated](https://rules.sonarsource.com/typescript/RSPEC-4158)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-4158)

["delete" should be used only with object properties](https://rules.sonarsource.com/typescript/RSPEC-3001)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-3001)

[Function parameters, caught exceptions and foreach variables' initial values should not be ignored](https://rules.sonarsource.com/typescript/RSPEC-1226)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-1226)

[Forwarding client IP address is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5759)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5759)

[Allowing confidential information to be logged is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5757)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5757)

[Allowing browsers to perform DNS prefetching is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5743)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5743)

[Disabling Certificate Transparency monitoring is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5742)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5742)

[Disabling Strict-Transport-Security policy is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5739)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5739)

[Disabling strict HTTP no-referrer policy is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5736)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5736)

[Allowing browsers to sniff MIME types is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5734)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5734)

[Disabling content security policy frame-ancestors directive is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5732)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5732)

[Allowing mixed-content is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5730)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5730)

[Disabling content security policy fetch directives is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5728)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5728)

[Disabling resource integrity features is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5725)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5725)

[Disclosing fingerprints from web application technologies is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5689)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5689)

[Having a permissive Cross-Origin Resource Sharing policy is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-5122)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-5122)

[Delivering code in production with debug features activated is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-4507)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-4507)

[Creating cookies without the "HttpOnly" flag is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-3330)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-3330)

[Creating cookies without the "secure" flag is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-2092)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-2092)

[Using hardcoded IP addresses is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-1313)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-1313)

[Regular expression quantifiers and character classes should be used concisely](https://rules.sonarsource.com/typescript/RSPEC-6353)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-6353)

[Regular expression literals should be used when possible](https://rules.sonarsource.com/typescript/RSPEC-6325)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-6325)

["await" should not be used redundantly](https://rules.sonarsource.com/typescript/RSPEC-4326)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4326)

[Redundant casts and non-null assertions should be avoided](https://rules.sonarsource.com/typescript/RSPEC-4325)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4325)

[Type aliases should be used](https://rules.sonarsource.com/typescript/RSPEC-4323)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4323)

[Type guards should be used](https://rules.sonarsource.com/typescript/RSPEC-4322)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4322)

["module" should not be used](https://rules.sonarsource.com/typescript/RSPEC-4156)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4156)

["for of" should be used with Iterables](https://rules.sonarsource.com/typescript/RSPEC-4138)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4138)

[Imports from the same modules should be merged](https://rules.sonarsource.com/typescript/RSPEC-3863)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3863)

[Jump statements should not be redundant](https://rules.sonarsource.com/typescript/RSPEC-3626)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3626)

[Default export names and file names should match](https://rules.sonarsource.com/typescript/RSPEC-3317)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3317)

[The global "this" object should not be used](https://rules.sonarsource.com/typescript/RSPEC-2990)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-2990)

["catch" clauses should do more than rethrow](https://rules.sonarsource.com/typescript/RSPEC-2737)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-2737)

[Boolean checks should not be inverted](https://rules.sonarsource.com/typescript/RSPEC-1940)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1940)

[Deprecated APIs should not be used](https://rules.sonarsource.com/typescript/RSPEC-1874)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1874)

[Wrapper objects should not be used for primitive types](https://rules.sonarsource.com/typescript/RSPEC-1533)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1533)

[Multiline string literals should not be used](https://rules.sonarsource.com/typescript/RSPEC-1516)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1516)

[Local variables should not be declared and then immediately returned or thrown](https://rules.sonarsource.com/typescript/RSPEC-1488)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1488)

[Function call arguments should not start on new lines](https://rules.sonarsource.com/typescript/RSPEC-1472)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1472)

["switch" statements should have at least 3 "case" clauses](https://rules.sonarsource.com/typescript/RSPEC-1301)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1301)

[A "while" loop should be used instead of a "for" loop](https://rules.sonarsource.com/typescript/RSPEC-1264)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1264)

[Unnecessary imports should be removed](https://rules.sonarsource.com/typescript/RSPEC-1128)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1128)

[Boolean literals should not be used in comparisons](https://rules.sonarsource.com/typescript/RSPEC-1125)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1125)

[Extra semicolons should be removed](https://rules.sonarsource.com/typescript/RSPEC-1116)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1116)

[Class names should comply with a naming convention](https://rules.sonarsource.com/typescript/RSPEC-101)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-101)

[Track uses of "TODO" tags](https://rules.sonarsource.com/typescript/RSPEC-1135)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1135)

[Web SQL databases should not be used](https://rules.sonarsource.com/typescript/RSPEC-2817)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-2817)

[Variables declared with "var" should be declared before they are used](https://rules.sonarsource.com/typescript/RSPEC-1526)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1526)

[Track lack of copyright and license headers](https://rules.sonarsource.com/typescript/RSPEC-1451)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1451)

[Reading the Standard Input is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-4829)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-4829)

[Using command line arguments is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-4823)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-4823)

[Using Sockets is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-4818)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-4818)

[Executing XPath expressions is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-4817)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-4817)

[Encrypting data is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-4787)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-4787)

[Using regular expressions is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-4784)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-4784)

[Class methods should be used instead of "prototype" assignments](https://rules.sonarsource.com/typescript/RSPEC-3525)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3525)

[Variables should be declared with "let" or "const"](https://rules.sonarsource.com/typescript/RSPEC-3504)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3504)

[Unchanged variables should be marked "const"](https://rules.sonarsource.com/typescript/RSPEC-3353)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3353)

[Wildcard imports should not be used](https://rules.sonarsource.com/typescript/RSPEC-2208)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-2208)

["switch" statements should not be nested](https://rules.sonarsource.com/typescript/RSPEC-1821)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1821)

[Cyclomatic Complexity of functions should not be too high](https://rules.sonarsource.com/typescript/RSPEC-1541)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1541)

["strict" mode should be used with caution](https://rules.sonarsource.com/typescript/RSPEC-1539)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1539)

[Control flow statements "if", "for", "while", "switch" and "try" should not be nested too deeply](https://rules.sonarsource.com/typescript/RSPEC-134)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-134)

["switch" statements should have "default" clauses](https://rules.sonarsource.com/typescript/RSPEC-131)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-131)

["if ... else if" constructs should end with "else" clauses](https://rules.sonarsource.com/typescript/RSPEC-126)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-126)

[Control structures should use curly braces](https://rules.sonarsource.com/typescript/RSPEC-121)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-121)

[String literals should not be duplicated](https://rules.sonarsource.com/typescript/RSPEC-1192)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1192)

[Expressions should not be too complex](https://rules.sonarsource.com/typescript/RSPEC-1067)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1067)

[Template literal placeholder syntax should not be used in regular strings](https://rules.sonarsource.com/typescript/RSPEC-3786)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-3786)

[Built-in objects should not be overridden](https://rules.sonarsource.com/typescript/RSPEC-2424)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-2424)

["for...in" loops should filter properties before acting on them](https://rules.sonarsource.com/typescript/RSPEC-1535)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-1535)

[Results of operations on strings should not be ignored](https://rules.sonarsource.com/typescript/RSPEC-1154)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-1154)

[Increment (++) and decrement (--) operators should not be used in a method call or mixed with other operators in an expression](https://rules.sonarsource.com/typescript/RSPEC-881)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-881)

[Optional boolean parameters should have default value](https://rules.sonarsource.com/typescript/RSPEC-4798)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4798)

[Union types should not have too many elements](https://rules.sonarsource.com/typescript/RSPEC-4622)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4622)

[Dependencies should be explicit](https://rules.sonarsource.com/typescript/RSPEC-4328)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4328)

["this" should not be assigned to variables](https://rules.sonarsource.com/typescript/RSPEC-4327)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4327)

[The "any" type should not be used](https://rules.sonarsource.com/typescript/RSPEC-4204)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4204)

["for in" should not be used with iterables](https://rules.sonarsource.com/typescript/RSPEC-4139)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4139)

[Functions should use "return" consistently](https://rules.sonarsource.com/typescript/RSPEC-3801)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3801)

["arguments" should not be accessed directly](https://rules.sonarsource.com/typescript/RSPEC-3513)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3513)

[Comparison operators should not be used with strings](https://rules.sonarsource.com/typescript/RSPEC-3003)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3003)

[Private properties that are only assigned in the constructor or at declaration should be "readonly"](https://rules.sonarsource.com/typescript/RSPEC-2933)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-2933)

[Property getters and setters should come in pairs](https://rules.sonarsource.com/typescript/RSPEC-2376)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-2376)

[JavaScript parser failure](https://rules.sonarsource.com/typescript/RSPEC-2260)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-2260)

[The ternary operator should not be used](https://rules.sonarsource.com/typescript/RSPEC-1774)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1774)

["===" and "!==" should be used instead of "==" and "!="](https://rules.sonarsource.com/typescript/RSPEC-1440)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1440)

[Functions should not have too many lines of code](https://rules.sonarsource.com/typescript/RSPEC-138)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-138)

[Track comments matching a regular expression](https://rules.sonarsource.com/typescript/RSPEC-124)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-124)

[Statements should be on separate lines](https://rules.sonarsource.com/typescript/RSPEC-122)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-122)

[Magic numbers should not be used](https://rules.sonarsource.com/typescript/RSPEC-109)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-109)

[Collapsible "if" statements should be merged](https://rules.sonarsource.com/typescript/RSPEC-1066)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1066)

[Standard outputs should not be used directly to log anything](https://rules.sonarsource.com/typescript/RSPEC-106)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-106)

[Files should not have too many lines of code](https://rules.sonarsource.com/typescript/RSPEC-104)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-104)

[Lines should not be too long](https://rules.sonarsource.com/typescript/RSPEC-103)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-103)

[Debugger statements should not be used](https://rules.sonarsource.com/typescript/RSPEC-1525)

[Vulnerability](https://rules.sonarsource.com/typescript/RSPEC-1525)

[Regular expressions using Unicode character classes or property escapes should enable the unicode flag](https://rules.sonarsource.com/typescript/RSPEC-5867)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-5867)

[The base should be provided to "parseInt"](https://rules.sonarsource.com/typescript/RSPEC-2427)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-2427)

[Function declarations should not be made within blocks](https://rules.sonarsource.com/typescript/RSPEC-1530)

[Bug](https://rules.sonarsource.com/typescript/RSPEC-1530)

[Writing cookies is security-sensitive](https://rules.sonarsource.com/typescript/RSPEC-2255)

[Security Hotspot](https://rules.sonarsource.com/typescript/RSPEC-2255)

["continue" should not be used](https://rules.sonarsource.com/typescript/RSPEC-909)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-909)

[Primitive return types should be used](https://rules.sonarsource.com/typescript/RSPEC-4324)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4324)

[Default type parameters should be omitted](https://rules.sonarsource.com/typescript/RSPEC-4157)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4157)

[Type assertions should use "as"](https://rules.sonarsource.com/typescript/RSPEC-4137)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4137)

[Method overloads should be grouped together](https://rules.sonarsource.com/typescript/RSPEC-4136)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4136)

[Interfaces should not be empty](https://rules.sonarsource.com/typescript/RSPEC-4023)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-4023)

[Trailing commas should be used](https://rules.sonarsource.com/typescript/RSPEC-3723)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3723)

["import" should be used to include external code](https://rules.sonarsource.com/typescript/RSPEC-3533)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3533)

[Braces and parentheses should be used consistently with arrow functions](https://rules.sonarsource.com/typescript/RSPEC-3524)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3524)

[Destructuring syntax should be used for assignments](https://rules.sonarsource.com/typescript/RSPEC-3514)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3514)

[Template strings should be used instead of concatenation](https://rules.sonarsource.com/typescript/RSPEC-3512)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3512)

[Shorthand object properties should be grouped at the beginning or end of an object declaration](https://rules.sonarsource.com/typescript/RSPEC-3499)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3499)

[Object literal shorthand syntax should be used](https://rules.sonarsource.com/typescript/RSPEC-3498)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3498)

[Strings and non-strings should not be added](https://rules.sonarsource.com/typescript/RSPEC-3402)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3402)

[Primitive types should be omitted from initialized or defaulted declarations](https://rules.sonarsource.com/typescript/RSPEC-3257)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-3257)

[Non-null assertions should not be used](https://rules.sonarsource.com/typescript/RSPEC-2966)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-2966)

["undefined" should not be assigned](https://rules.sonarsource.com/typescript/RSPEC-2138)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-2138)

[Trailing commas should not be used](https://rules.sonarsource.com/typescript/RSPEC-1537)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1537)

[Array constructors should not be used](https://rules.sonarsource.com/typescript/RSPEC-1528)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1528)

[Quotes for string literals should be used consistently](https://rules.sonarsource.com/typescript/RSPEC-1441)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1441)

[Statements should end with semicolons](https://rules.sonarsource.com/typescript/RSPEC-1438)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1438)

[Comments should not be located at the end of lines of code](https://rules.sonarsource.com/typescript/RSPEC-139)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-139)

[Loops should not contain more than a single "break" or "continue" statement](https://rules.sonarsource.com/typescript/RSPEC-135)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-135)

[Variable, property and parameter names should comply with a naming convention](https://rules.sonarsource.com/typescript/RSPEC-117)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-117)

[Lines should not end with trailing whitespaces](https://rules.sonarsource.com/typescript/RSPEC-1131)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1131)

[Files should contain an empty newline at the end](https://rules.sonarsource.com/typescript/RSPEC-113)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-113)

[An open curly brace should be located at the end of a line](https://rules.sonarsource.com/typescript/RSPEC-1105)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-1105)

[Tabulation characters should not be used](https://rules.sonarsource.com/typescript/RSPEC-105)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-105)

[Function and method names should comply with a naming convention](https://rules.sonarsource.com/typescript/RSPEC-100)

[Code Smell](https://rules.sonarsource.com/typescript/RSPEC-100)