## JavaScript static code analysis

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### [Callbacks of array methods should have return statements](https://rules.sonarsource.com/javascript/RSPEC-3796)

[Bug](https://rules.sonarsource.com/javascript/RSPEC-3796)

### [Loops should not be infinite](https://rules.sonarsource.com/javascript/RSPEC-2189)

[Bug](https://rules.sonarsource.com/javascript/RSPEC-2189)

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

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

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

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

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

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

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

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

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

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

### [Variables should be declared explicitly](https://rules.sonarsource.com/javascript/RSPEC-2703)

[Code Smell](https://rules.sonarsource.com/javascript/RSPEC-2703)

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

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

### ["future reserved words" should not be used as identifiers](https://rules.sonarsource.com/javascript/RSPEC-1527)

[Code Smell](https://rules.sonarsource.com/javascript/RSPEC-1527)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### [Function calls should not pass extra arguments](https://rules.sonarsource.com/javascript/RSPEC-930)

[Bug](https://rules.sonarsource.com/javascript/RSPEC-930)

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

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

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

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

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

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

### ["Symbol" should not be used as a constructor](https://rules.sonarsource.com/javascript/RSPEC-3834)

[Bug](https://rules.sonarsource.com/javascript/RSPEC-3834)

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

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

### ["in" should not be used with primitive types](https://rules.sonarsource.com/javascript/RSPEC-3785)

[Bug](https://rules.sonarsource.com/javascript/RSPEC-3785)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### [Attempts should not be made to update "const" variables](https://rules.sonarsource.com/javascript/RSPEC-3500)

[Bug](https://rules.sonarsource.com/javascript/RSPEC-3500)

### [Strict equality operators should not be used with dissimilar types](https://rules.sonarsource.com/javascript/RSPEC-3403)

[Bug](https://rules.sonarsource.com/javascript/RSPEC-3403)

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

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

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

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

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

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

### [Setters should not return values](https://rules.sonarsource.com/javascript/RSPEC-2432)

[Bug](https://rules.sonarsource.com/javascript/RSPEC-2432)

### [Properties of variables with "null" or "undefined" values should not be accessed](https://rules.sonarsource.com/javascript/RSPEC-2259)

[Bug](https://rules.sonarsource.com/javascript/RSPEC-2259)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### [Function argument names should be unique](https://rules.sonarsource.com/javascript/RSPEC-1536)

[Bug](https://rules.sonarsource.com/javascript/RSPEC-1536)

### [Property names should not be duplicated within a class or object literal](https://rules.sonarsource.com/javascript/RSPEC-1534)

[Bug](https://rules.sonarsource.com/javascript/RSPEC-1534)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### [Functions should always return the same type](https://rules.sonarsource.com/javascript/RSPEC-3800)

[Code Smell](https://rules.sonarsource.com/javascript/RSPEC-3800)

### [Arguments to built-in functions should match documented types](https://rules.sonarsource.com/javascript/RSPEC-3782)

[Code Smell](https://rules.sonarsource.com/javascript/RSPEC-3782)

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

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

### [Functions should not be called both with and without "new"](https://rules.sonarsource.com/javascript/RSPEC-3686)

[Code Smell](https://rules.sonarsource.com/javascript/RSPEC-3686)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### ["with" statements should not be used](https://rules.sonarsource.com/javascript/RSPEC-1321)

[Bug](https://rules.sonarsource.com/javascript/RSPEC-1321)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### [Unused local variables and functions should be removed](https://rules.sonarsource.com/javascript/RSPEC-1481)

[Code Smell](https://rules.sonarsource.com/javascript/RSPEC-1481)

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

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

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

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

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

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

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

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

### [Return of boolean expressions should not be wrapped into an "if-then-else" statement](https://rules.sonarsource.com/javascript/RSPEC-1126)

[Code Smell](https://rules.sonarsource.com/javascript/RSPEC-1126)

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

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

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

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

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

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

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

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

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

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

### [Variables should be defined before being used](https://rules.sonarsource.com/javascript/RSPEC-3827)

[Bug](https://rules.sonarsource.com/javascript/RSPEC-3827)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### [Function constructors should not be used](https://rules.sonarsource.com/javascript/RSPEC-3523)

[Code Smell](https://rules.sonarsource.com/javascript/RSPEC-3523)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### [Local storage should not be used](https://rules.sonarsource.com/javascript/RSPEC-3271)

[Vulnerability](https://rules.sonarsource.com/javascript/RSPEC-3271)

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

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

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

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

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

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

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

[Bug](https://rules.sonarsource.com/javascript/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/javascript/RSPEC-881)

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

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

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

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

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

### [Variables and functions should not be declared in the global scope](https://rules.sonarsource.com/javascript/RSPEC-3798)

[Code Smell](https://rules.sonarsource.com/javascript/RSPEC-3798)

### [Arithmetic operators should only have numbers as operands](https://rules.sonarsource.com/javascript/RSPEC-3760)

[Code Smell](https://rules.sonarsource.com/javascript/RSPEC-3760)

### [Values not convertible to numbers should not be used in numeric comparisons](https://rules.sonarsource.com/javascript/RSPEC-3758)

[Code Smell](https://rules.sonarsource.com/javascript/RSPEC-3758)

### [Arithmetic operations should not result in "NaN"](https://rules.sonarsource.com/javascript/RSPEC-3757)

[Code Smell](https://rules.sonarsource.com/javascript/RSPEC-3757)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### ["alert(...)" should not be used](https://rules.sonarsource.com/javascript/RSPEC-1442)

[Vulnerability](https://rules.sonarsource.com/javascript/RSPEC-1442)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

### [Object literal syntax should be used](https://rules.sonarsource.com/javascript/RSPEC-2428)

[Code Smell](https://rules.sonarsource.com/javascript/RSPEC-2428)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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