## Java static code analysis

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

### [Components should not be vulnerable to intent redirection](https://rules.sonarsource.com/java/RSPEC-6384)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-6384)

### [XML parsers should not allow inclusion of arbitrary files](https://rules.sonarsource.com/java/RSPEC-6373)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-6373)

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

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

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

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

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

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

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

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

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

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

### [Deserialization should not be vulnerable to injection attacks](https://rules.sonarsource.com/java/RSPEC-5135)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-5135)

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

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

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

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

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

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

### [A secure password should be used when connecting to a database](https://rules.sonarsource.com/java/RSPEC-2115)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-2115)

### [XPath expressions should not be vulnerable to injection attacks](https://rules.sonarsource.com/java/RSPEC-2091)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-2091)

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

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

### [LDAP queries should not be vulnerable to injection attacks](https://rules.sonarsource.com/java/RSPEC-2078)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-2078)

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

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

### ["@SpringBootApplication" and "@ComponentScan" should not be used in the default package](https://rules.sonarsource.com/java/RSPEC-4602)

[Bug](https://rules.sonarsource.com/java/RSPEC-4602)

### ["@Controller" classes that use "@SessionAttributes" must call "setComplete" on their "SessionStatus" objects](https://rules.sonarsource.com/java/RSPEC-3753)

[Bug](https://rules.sonarsource.com/java/RSPEC-3753)

### ["wait" should not be called when multiple locks are held](https://rules.sonarsource.com/java/RSPEC-3046)

[Bug](https://rules.sonarsource.com/java/RSPEC-3046)

### ["PreparedStatement" and "ResultSet" methods should be called with valid indices](https://rules.sonarsource.com/java/RSPEC-2695)

[Bug](https://rules.sonarsource.com/java/RSPEC-2695)

### [Files opened in append mode should not be used with ObjectOutputStream](https://rules.sonarsource.com/java/RSPEC-2689)

[Bug](https://rules.sonarsource.com/java/RSPEC-2689)

### ["wait(...)" should be used instead of "Thread.sleep(...)" when a lock is held](https://rules.sonarsource.com/java/RSPEC-2276)

[Bug](https://rules.sonarsource.com/java/RSPEC-2276)

### [Printf-style format strings should not lead to unexpected behavior at runtime](https://rules.sonarsource.com/java/RSPEC-2275)

[Bug](https://rules.sonarsource.com/java/RSPEC-2275)

### [Methods "wait(...)", "notify()" and "notifyAll()" should not be called on Thread instances](https://rules.sonarsource.com/java/RSPEC-2236)

[Bug](https://rules.sonarsource.com/java/RSPEC-2236)

### [Methods should not call same-class methods with incompatible "@Transactional" values](https://rules.sonarsource.com/java/RSPEC-2229)

[Bug](https://rules.sonarsource.com/java/RSPEC-2229)

### [Recursion should not be infinite](https://rules.sonarsource.com/java/RSPEC-2190)

[Bug](https://rules.sonarsource.com/java/RSPEC-2190)

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

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

### [Double-checked locking should not be used](https://rules.sonarsource.com/java/RSPEC-2168)

[Bug](https://rules.sonarsource.com/java/RSPEC-2168)

### [Resources should be closed](https://rules.sonarsource.com/java/RSPEC-2095)

[Bug](https://rules.sonarsource.com/java/RSPEC-2095)

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

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

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

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

### ["ThreadGroup" should not be used](https://rules.sonarsource.com/java/RSPEC-3014)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3014)

### ["clone" should not be overridden](https://rules.sonarsource.com/java/RSPEC-2975)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2975)

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

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

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

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

### [Silly bit operations should not be performed](https://rules.sonarsource.com/java/RSPEC-2437)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2437)

### [Child class fields should not shadow parent class fields](https://rules.sonarsource.com/java/RSPEC-2387)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2387)

### [JUnit test cases should call super methods](https://rules.sonarsource.com/java/RSPEC-2188)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2188)

### [TestCases should contain tests](https://rules.sonarsource.com/java/RSPEC-2187)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2187)

### [Short-circuit logic should be used in boolean contexts](https://rules.sonarsource.com/java/RSPEC-2178)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2178)

### [Methods and field names should not be the same or differ only by capitalization](https://rules.sonarsource.com/java/RSPEC-1845)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1845)

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

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

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

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

### [Future keywords should not be used as names](https://rules.sonarsource.com/java/RSPEC-1190)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1190)

### [Thread suspensions should not be vulnerable to Denial of Service attacks](https://rules.sonarsource.com/java/RSPEC-6390)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-6390)

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

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

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

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

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

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

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

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

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

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

### [Insecure temporary file creation methods should not be used](https://rules.sonarsource.com/java/RSPEC-5445)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-5445)

### [Passwords should not be stored in plain-text or with a fast hashing algorithm](https://rules.sonarsource.com/java/RSPEC-5344)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-5344)

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

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

### [Persistent entities should not be used as arguments of "@RequestMapping" methods](https://rules.sonarsource.com/java/RSPEC-4684)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-4684)

### ["HttpSecurity" URL patterns should be correctly ordered](https://rules.sonarsource.com/java/RSPEC-4601)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-4601)

### [LDAP connections should be authenticated](https://rules.sonarsource.com/java/RSPEC-4433)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-4433)

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

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

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

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

### ["SecureRandom" seeds should not be predictable](https://rules.sonarsource.com/java/RSPEC-4347)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-4347)

### [Cipher Block Chaining IVs should be unpredictable](https://rules.sonarsource.com/java/RSPEC-3329)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-3329)

### [Basic authentication should not be used](https://rules.sonarsource.com/java/RSPEC-2647)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-2647)

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

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

### ["HttpServletRequest.getRequestedSessionId()" should not be used](https://rules.sonarsource.com/java/RSPEC-2254)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-2254)

### [Hashes should include an unpredictable salt](https://rules.sonarsource.com/java/RSPEC-2053)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-2053)

### [Calls to methods should not trigger an IllegalArgumentException](https://rules.sonarsource.com/java/RSPEC-6416)

[Bug](https://rules.sonarsource.com/java/RSPEC-6416)

### [Unsupported methods should not be called on some collection implementations](https://rules.sonarsource.com/java/RSPEC-6322)

[Bug](https://rules.sonarsource.com/java/RSPEC-6322)

### [Cast operations should not trigger a ClassCastException](https://rules.sonarsource.com/java/RSPEC-6320)

[Bug](https://rules.sonarsource.com/java/RSPEC-6320)

### [Members ignored during record serialization should not be used](https://rules.sonarsource.com/java/RSPEC-6209)

[Bug](https://rules.sonarsource.com/java/RSPEC-6209)

### [Map "computeIfAbsent()" and "computeIfPresent()" should not be used to add "null" values.](https://rules.sonarsource.com/java/RSPEC-6104)

[Bug](https://rules.sonarsource.com/java/RSPEC-6104)

### [Regex lookahead assertions should not be contradictory](https://rules.sonarsource.com/java/RSPEC-6002)

[Bug](https://rules.sonarsource.com/java/RSPEC-6002)

### [Back references in regular expressions should only refer to capturing groups that are matched before the reference](https://rules.sonarsource.com/java/RSPEC-6001)

[Bug](https://rules.sonarsource.com/java/RSPEC-6001)

### [Regex boundaries should not be used in a way that can never be matched](https://rules.sonarsource.com/java/RSPEC-5996)

[Bug](https://rules.sonarsource.com/java/RSPEC-5996)

### [Regex patterns following a possessive quantifier should not always fail](https://rules.sonarsource.com/java/RSPEC-5994)

[Bug](https://rules.sonarsource.com/java/RSPEC-5994)

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

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

### [Assertions comparing incompatible types should not be made](https://rules.sonarsource.com/java/RSPEC-5845)

[Bug](https://rules.sonarsource.com/java/RSPEC-5845)

### [JUnit5 inner test classes should be annotated with @Nested](https://rules.sonarsource.com/java/RSPEC-5790)

[Bug](https://rules.sonarsource.com/java/RSPEC-5790)

### [Only one method invocation is expected when testing checked exceptions](https://rules.sonarsource.com/java/RSPEC-5783)

[Bug](https://rules.sonarsource.com/java/RSPEC-5783)

### [Assertion methods should not be used within the try block of a try-catch catching an Error](https://rules.sonarsource.com/java/RSPEC-5779)

[Bug](https://rules.sonarsource.com/java/RSPEC-5779)

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

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

### [Zero should not be a possible denominator](https://rules.sonarsource.com/java/RSPEC-3518)

[Bug](https://rules.sonarsource.com/java/RSPEC-3518)

### [Locks should be released](https://rules.sonarsource.com/java/RSPEC-2222)

[Bug](https://rules.sonarsource.com/java/RSPEC-2222)

### ["runFinalizersOnExit" should not be called](https://rules.sonarsource.com/java/RSPEC-2151)

[Bug](https://rules.sonarsource.com/java/RSPEC-2151)

### ["ScheduledThreadPoolExecutor" should not have 0 core threads](https://rules.sonarsource.com/java/RSPEC-2122)

[Bug](https://rules.sonarsource.com/java/RSPEC-2122)

### ["Random" objects should be reused](https://rules.sonarsource.com/java/RSPEC-2119)

[Bug](https://rules.sonarsource.com/java/RSPEC-2119)

### [The signature of "finalize()" should match that of "Object.finalize()"](https://rules.sonarsource.com/java/RSPEC-1175)

[Bug](https://rules.sonarsource.com/java/RSPEC-1175)

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

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

### ["super.finalize()" should be called at the end of "Object.finalize()" implementations](https://rules.sonarsource.com/java/RSPEC-1114)

[Bug](https://rules.sonarsource.com/java/RSPEC-1114)

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

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

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

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

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

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

### [Accessing Android external storage is security-sensitive](https://rules.sonarsource.com/java/RSPEC-5324)

[Security Hotspot](https://rules.sonarsource.com/java/RSPEC-5324)

### [Receiving intents is security-sensitive](https://rules.sonarsource.com/java/RSPEC-5322)

[Security Hotspot](https://rules.sonarsource.com/java/RSPEC-5322)

### [Broadcasting intents is security-sensitive](https://rules.sonarsource.com/java/RSPEC-5320)

[Security Hotspot](https://rules.sonarsource.com/java/RSPEC-5320)

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

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

### [Configuring loggers is security-sensitive](https://rules.sonarsource.com/java/RSPEC-4792)

[Security Hotspot](https://rules.sonarsource.com/java/RSPEC-4792)

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

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

### [Using unsafe Jackson deserialization configuration is security-sensitive](https://rules.sonarsource.com/java/RSPEC-4544)

[Security Hotspot](https://rules.sonarsource.com/java/RSPEC-4544)

### [Setting JavaBean properties is security-sensitive](https://rules.sonarsource.com/java/RSPEC-4512)

[Security Hotspot](https://rules.sonarsource.com/java/RSPEC-4512)

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

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

### [Using non-standard cryptographic algorithms is security-sensitive](https://rules.sonarsource.com/java/RSPEC-2257)

[Security Hotspot](https://rules.sonarsource.com/java/RSPEC-2257)

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

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

### [Mocking all non-private methods of a class should be avoided](https://rules.sonarsource.com/java/RSPEC-5969)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5969)

### [Empty lines should not be tested with regex MULTILINE flag](https://rules.sonarsource.com/java/RSPEC-5846)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5846)

### [Methods setUp() and tearDown() should be correctly annotated starting with JUnit4](https://rules.sonarsource.com/java/RSPEC-5826)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5826)

### [Class members annotated with "@VisibleForTesting" should not be accessed from production code](https://rules.sonarsource.com/java/RSPEC-5803)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5803)

### ["String#replace" should be preferred to "String#replaceAll"](https://rules.sonarsource.com/java/RSPEC-5361)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5361)

### [Derived exceptions should not hide their parents' catch blocks](https://rules.sonarsource.com/java/RSPEC-4970)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4970)

### [String offset-based methods should be preferred for finding substrings from offsets](https://rules.sonarsource.com/java/RSPEC-4635)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4635)

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

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

### ["equals" method parameters should not be marked "@Nonnull"](https://rules.sonarsource.com/java/RSPEC-4454)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4454)

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

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

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

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

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

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

### [Factory method injection should be used in "@Configuration" classes](https://rules.sonarsource.com/java/RSPEC-3305)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3305)

### ["static" base class members should not be accessed via derived types](https://rules.sonarsource.com/java/RSPEC-3252)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3252)

### [Instance methods should not write to "static" fields](https://rules.sonarsource.com/java/RSPEC-2696)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2696)

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

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

### [Method overrides should not change contracts](https://rules.sonarsource.com/java/RSPEC-2638)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2638)

### [Whitespace and control characters in literals should be explicit](https://rules.sonarsource.com/java/RSPEC-2479)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2479)

### [Null should not be returned from a "Boolean" method](https://rules.sonarsource.com/java/RSPEC-2447)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2447)

### [Classes should not access their own subclasses during initialization](https://rules.sonarsource.com/java/RSPEC-2390)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2390)

### ["Object.wait(...)" and "Condition.await(...)" should be called inside a "while" loop](https://rules.sonarsource.com/java/RSPEC-2274)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2274)

### [IllegalMonitorStateException should not be caught](https://rules.sonarsource.com/java/RSPEC-2235)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2235)

### [JUnit assertions should not be used in "run" methods](https://rules.sonarsource.com/java/RSPEC-2186)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2186)

### [Class names should not shadow interfaces or superclasses](https://rules.sonarsource.com/java/RSPEC-2176)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2176)

### ["Cloneables" should implement "clone"](https://rules.sonarsource.com/java/RSPEC-2157)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2157)

### [Try-with-resources should be used](https://rules.sonarsource.com/java/RSPEC-2093)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2093)

### ["readResolve" methods should be inheritable](https://rules.sonarsource.com/java/RSPEC-2062)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2062)

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

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

### [Fields in a "Serializable" class should either be transient or serializable](https://rules.sonarsource.com/java/RSPEC-1948)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1948)

### [Package declaration should match source file directory](https://rules.sonarsource.com/java/RSPEC-1598)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1598)

### [Generic wildcard types should not be used in return types](https://rules.sonarsource.com/java/RSPEC-1452)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1452)

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

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

### [Execution of the Garbage Collector should be triggered only by the JVM](https://rules.sonarsource.com/java/RSPEC-1215)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1215)

### [Constants should not be defined in interfaces](https://rules.sonarsource.com/java/RSPEC-1214)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1214)

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

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

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

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

### ["Object.finalize()" should remain protected (versus public) when overriding](https://rules.sonarsource.com/java/RSPEC-1174)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1174)

### [Exceptions should not be thrown in finally blocks](https://rules.sonarsource.com/java/RSPEC-1163)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1163)

### [Constant names should comply with a naming convention](https://rules.sonarsource.com/java/RSPEC-115)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-115)

### [The Object.finalize() method should not be overridden](https://rules.sonarsource.com/java/RSPEC-1113)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1113)

### [XML operations should not be vulnerable to injection attacks](https://rules.sonarsource.com/java/RSPEC-6399)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-6399)

### [JSON operations should not be vulnerable to injection attacks](https://rules.sonarsource.com/java/RSPEC-6398)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-6398)

### [XML signatures should be validated securely](https://rules.sonarsource.com/java/RSPEC-6377)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-6377)

### [XML parsers should not be vulnerable to Denial of Service attacks](https://rules.sonarsource.com/java/RSPEC-6376)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-6376)

### [XML parsers should not load external schemas](https://rules.sonarsource.com/java/RSPEC-6374)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-6374)

### [Mobile database encryption keys should not be disclosed](https://rules.sonarsource.com/java/RSPEC-6301)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-6301)

### [Reflection should not be vulnerable to injection attacks](https://rules.sonarsource.com/java/RSPEC-6173)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-6173)

### [Authorizations should be based on strong decisions](https://rules.sonarsource.com/java/RSPEC-5808)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-5808)

### [OpenSAML2 should be configured to prevent authentication bypass](https://rules.sonarsource.com/java/RSPEC-5679)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-5679)

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

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

### [Collections should not be modified while they are iterated](https://rules.sonarsource.com/java/RSPEC-6417)

[Bug](https://rules.sonarsource.com/java/RSPEC-6417)

### [Equals method should be overridden in records containing array fields](https://rules.sonarsource.com/java/RSPEC-6218)

[Bug](https://rules.sonarsource.com/java/RSPEC-6218)

### [Reflection should not be used to increase accessibility of records' fields](https://rules.sonarsource.com/java/RSPEC-6216)

[Bug](https://rules.sonarsource.com/java/RSPEC-6216)

### [AssertJ assertions with "Consumer" arguments should contain assertion inside consumers](https://rules.sonarsource.com/java/RSPEC-6103)

[Bug](https://rules.sonarsource.com/java/RSPEC-6103)

### [The regex escape sequence \cX should only be used with characters in the @-\_ range](https://rules.sonarsource.com/java/RSPEC-6070)

[Bug](https://rules.sonarsource.com/java/RSPEC-6070)

### [Regular expressions should not overflow the stack](https://rules.sonarsource.com/java/RSPEC-5998)

[Bug](https://rules.sonarsource.com/java/RSPEC-5998)

### [Tests method should not be annotated with competing annotations](https://rules.sonarsource.com/java/RSPEC-5967)

[Bug](https://rules.sonarsource.com/java/RSPEC-5967)

### [Assertions should not be used in production code](https://rules.sonarsource.com/java/RSPEC-5960)

[Bug](https://rules.sonarsource.com/java/RSPEC-5960)

### [DateTimeFormatters should not use mismatched year and week numbers](https://rules.sonarsource.com/java/RSPEC-5917)

[Bug](https://rules.sonarsource.com/java/RSPEC-5917)

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

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

### [Case insensitive Unicode regular expressions should enable the “UNICODE\_CASE” flag](https://rules.sonarsource.com/java/RSPEC-5866)

[Bug](https://rules.sonarsource.com/java/RSPEC-5866)

### [Assertions should not compare an object to itself](https://rules.sonarsource.com/java/RSPEC-5863)

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

### [Regex alternatives should not be redundant](https://rules.sonarsource.com/java/RSPEC-5855)

[Bug](https://rules.sonarsource.com/java/RSPEC-5855)

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

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

### [AssertJ methods setting the assertion context should come before an assertion](https://rules.sonarsource.com/java/RSPEC-5833)

[Bug](https://rules.sonarsource.com/java/RSPEC-5833)

### [AssertJ configuration should be applied](https://rules.sonarsource.com/java/RSPEC-5831)

[Bug](https://rules.sonarsource.com/java/RSPEC-5831)

### [JUnit5 test classes and methods should not be silently ignored](https://rules.sonarsource.com/java/RSPEC-5810)

[Bug](https://rules.sonarsource.com/java/RSPEC-5810)

### ["ThreadLocal" variables should be cleaned up when no longer used](https://rules.sonarsource.com/java/RSPEC-5164)

[Bug](https://rules.sonarsource.com/java/RSPEC-5164)

### [Strings and Boxed types should be compared using "equals()"](https://rules.sonarsource.com/java/RSPEC-4973)

[Bug](https://rules.sonarsource.com/java/RSPEC-4973)

### [InputSteam.read() implementation should not return a signed byte](https://rules.sonarsource.com/java/RSPEC-4517)

[Bug](https://rules.sonarsource.com/java/RSPEC-4517)

### ["compareTo" should not be overloaded](https://rules.sonarsource.com/java/RSPEC-4351)

[Bug](https://rules.sonarsource.com/java/RSPEC-4351)

### ["iterator" should not return "this"](https://rules.sonarsource.com/java/RSPEC-4348)

[Bug](https://rules.sonarsource.com/java/RSPEC-4348)

### [Map values should not be replaced unconditionally](https://rules.sonarsource.com/java/RSPEC-4143)

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

### [Week Year ("YYYY") should not be used for date formatting](https://rules.sonarsource.com/java/RSPEC-3986)

[Bug](https://rules.sonarsource.com/java/RSPEC-3986)

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

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

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

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

### [Consumed Stream pipelines should not be reused](https://rules.sonarsource.com/java/RSPEC-3959)

[Bug](https://rules.sonarsource.com/java/RSPEC-3959)

### [Intermediate Stream methods should not be left unused](https://rules.sonarsource.com/java/RSPEC-3958)

[Bug](https://rules.sonarsource.com/java/RSPEC-3958)

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

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

### [Optional value should only be accessed after calling isPresent()](https://rules.sonarsource.com/java/RSPEC-3655)

[Bug](https://rules.sonarsource.com/java/RSPEC-3655)

### [Overrides should match their parent class methods in synchronization](https://rules.sonarsource.com/java/RSPEC-3551)

[Bug](https://rules.sonarsource.com/java/RSPEC-3551)

### [Value-based classes should not be used for locking](https://rules.sonarsource.com/java/RSPEC-3436)

[Bug](https://rules.sonarsource.com/java/RSPEC-3436)

### [Expressions used in "assert" should not produce side effects](https://rules.sonarsource.com/java/RSPEC-3346)

[Bug](https://rules.sonarsource.com/java/RSPEC-3346)

### ["volatile" variables should not be used with compound operators](https://rules.sonarsource.com/java/RSPEC-3078)

[Bug](https://rules.sonarsource.com/java/RSPEC-3078)

### ["getClass" should not be used for synchronization](https://rules.sonarsource.com/java/RSPEC-3067)

[Bug](https://rules.sonarsource.com/java/RSPEC-3067)

### [Min and max used in combination should not always return the same value](https://rules.sonarsource.com/java/RSPEC-3065)

[Bug](https://rules.sonarsource.com/java/RSPEC-3065)

### [Assignment of lazy-initialized members should be the last step with double-checked locking](https://rules.sonarsource.com/java/RSPEC-3064)

[Bug](https://rules.sonarsource.com/java/RSPEC-3064)

### ["String" calls should not go beyond their bounds](https://rules.sonarsource.com/java/RSPEC-3039)

[Bug](https://rules.sonarsource.com/java/RSPEC-3039)

### [Raw byte values should not be used in bitwise operations in combination with shifts](https://rules.sonarsource.com/java/RSPEC-3034)

[Bug](https://rules.sonarsource.com/java/RSPEC-3034)

### [Getters and setters should be synchronized in pairs](https://rules.sonarsource.com/java/RSPEC-2886)

[Bug](https://rules.sonarsource.com/java/RSPEC-2886)

### [Non-thread-safe fields should not be static](https://rules.sonarsource.com/java/RSPEC-2885)

[Bug](https://rules.sonarsource.com/java/RSPEC-2885)

### ["null" should not be used with "Optional"](https://rules.sonarsource.com/java/RSPEC-2789)

[Bug](https://rules.sonarsource.com/java/RSPEC-2789)

### [Unary prefix operators should not be repeated](https://rules.sonarsource.com/java/RSPEC-2761)

[Bug](https://rules.sonarsource.com/java/RSPEC-2761)

### ["=+" should not be used instead of "+="](https://rules.sonarsource.com/java/RSPEC-2757)

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

### ["read" and "readLine" return values should be used](https://rules.sonarsource.com/java/RSPEC-2677)

[Bug](https://rules.sonarsource.com/java/RSPEC-2677)

### [Inappropriate regular expressions should not be used](https://rules.sonarsource.com/java/RSPEC-2639)

[Bug](https://rules.sonarsource.com/java/RSPEC-2639)

### [Conditionally executed code should be reachable](https://rules.sonarsource.com/java/RSPEC-2583)

[Bug](https://rules.sonarsource.com/java/RSPEC-2583)

### ["notifyAll" should be used](https://rules.sonarsource.com/java/RSPEC-2446)

[Bug](https://rules.sonarsource.com/java/RSPEC-2446)

### [Blocks should be synchronized on "private final" fields](https://rules.sonarsource.com/java/RSPEC-2445)

[Bug](https://rules.sonarsource.com/java/RSPEC-2445)

### [Non-serializable objects should not be stored in "HttpSession" objects](https://rules.sonarsource.com/java/RSPEC-2441)

[Bug](https://rules.sonarsource.com/java/RSPEC-2441)

### ["wait", "notify" and "notifyAll" should only be called when a lock is obviously held on an object](https://rules.sonarsource.com/java/RSPEC-2273)

[Bug](https://rules.sonarsource.com/java/RSPEC-2273)

### [Null pointers should not be dereferenced](https://rules.sonarsource.com/java/RSPEC-2259)

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

### [Loop conditions should be true at least once](https://rules.sonarsource.com/java/RSPEC-2252)

[Bug](https://rules.sonarsource.com/java/RSPEC-2252)

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

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

### [Non-public methods should not be "@Transactional"](https://rules.sonarsource.com/java/RSPEC-2230)

[Bug](https://rules.sonarsource.com/java/RSPEC-2230)

### [Servlets should not have mutable instance fields](https://rules.sonarsource.com/java/RSPEC-2226)

[Bug](https://rules.sonarsource.com/java/RSPEC-2226)

### ["toString()" and "clone()" methods should not return null](https://rules.sonarsource.com/java/RSPEC-2225)

[Bug](https://rules.sonarsource.com/java/RSPEC-2225)

### [".equals()" should not be used to test the values of "Atomic" classes](https://rules.sonarsource.com/java/RSPEC-2204)

[Bug](https://rules.sonarsource.com/java/RSPEC-2204)

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

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

### [Child class methods named for parent class methods should be overrides](https://rules.sonarsource.com/java/RSPEC-2177)

[Bug](https://rules.sonarsource.com/java/RSPEC-2177)

### [Inappropriate "Collection" calls should not be made](https://rules.sonarsource.com/java/RSPEC-2175)

[Bug](https://rules.sonarsource.com/java/RSPEC-2175)

### [Silly equality checks should not be made](https://rules.sonarsource.com/java/RSPEC-2159)

[Bug](https://rules.sonarsource.com/java/RSPEC-2159)

### [Dissimilar primitive wrappers should not be used with the ternary operator without explicit casting](https://rules.sonarsource.com/java/RSPEC-2154)

[Bug](https://rules.sonarsource.com/java/RSPEC-2154)

### ["InterruptedException" should not be ignored](https://rules.sonarsource.com/java/RSPEC-2142)

[Bug](https://rules.sonarsource.com/java/RSPEC-2142)

### [Classes extending java.lang.Thread should override the "run" method](https://rules.sonarsource.com/java/RSPEC-2134)

[Bug](https://rules.sonarsource.com/java/RSPEC-2134)

### ["Double.longBitsToDouble" should not be used for "int"](https://rules.sonarsource.com/java/RSPEC-2127)

[Bug](https://rules.sonarsource.com/java/RSPEC-2127)

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

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

### [Silly String operations should not be made](https://rules.sonarsource.com/java/RSPEC-2121)

[Bug](https://rules.sonarsource.com/java/RSPEC-2121)

### [Non-serializable classes should not be written](https://rules.sonarsource.com/java/RSPEC-2118)

[Bug](https://rules.sonarsource.com/java/RSPEC-2118)

### ["hashCode" and "toString" should not be called on array instances](https://rules.sonarsource.com/java/RSPEC-2116)

[Bug](https://rules.sonarsource.com/java/RSPEC-2116)

### [Collections should not be passed as arguments to their own methods](https://rules.sonarsource.com/java/RSPEC-2114)

[Bug](https://rules.sonarsource.com/java/RSPEC-2114)

### ["BigDecimal(double)" should not be used](https://rules.sonarsource.com/java/RSPEC-2111)

[Bug](https://rules.sonarsource.com/java/RSPEC-2111)

### [Invalid "Date" values should not be used](https://rules.sonarsource.com/java/RSPEC-2110)

[Bug](https://rules.sonarsource.com/java/RSPEC-2110)

### [Reflection should not be used to check non-runtime annotations](https://rules.sonarsource.com/java/RSPEC-2109)

[Bug](https://rules.sonarsource.com/java/RSPEC-2109)

### [Custom serialization method signatures should meet requirements](https://rules.sonarsource.com/java/RSPEC-2061)

[Bug](https://rules.sonarsource.com/java/RSPEC-2061)

### ["Externalizable" classes should have no-arguments constructors](https://rules.sonarsource.com/java/RSPEC-2060)

[Bug](https://rules.sonarsource.com/java/RSPEC-2060)

### [Classes should not be compared by name](https://rules.sonarsource.com/java/RSPEC-1872)

[Bug](https://rules.sonarsource.com/java/RSPEC-1872)

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

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

### [Synchronization should not be done on instances of value-based classes](https://rules.sonarsource.com/java/RSPEC-1860)

[Bug](https://rules.sonarsource.com/java/RSPEC-1860)

### ["Iterator.hasNext()" should not call "Iterator.next()"](https://rules.sonarsource.com/java/RSPEC-1849)

[Bug](https://rules.sonarsource.com/java/RSPEC-1849)

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

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

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

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

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

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

### ["StringBuilder" and "StringBuffer" should not be instantiated with a character](https://rules.sonarsource.com/java/RSPEC-1317)

[Bug](https://rules.sonarsource.com/java/RSPEC-1317)

### [Methods should not be named "tostring", "hashcode" or "equal"](https://rules.sonarsource.com/java/RSPEC-1221)

[Bug](https://rules.sonarsource.com/java/RSPEC-1221)

### ["Thread.run()" should not be called directly](https://rules.sonarsource.com/java/RSPEC-1217)

[Bug](https://rules.sonarsource.com/java/RSPEC-1217)

### ["equals" method overrides should accept "Object" parameters](https://rules.sonarsource.com/java/RSPEC-1201)

[Bug](https://rules.sonarsource.com/java/RSPEC-1201)

### [The Object.finalize() method should not be called](https://rules.sonarsource.com/java/RSPEC-1111)

[Bug](https://rules.sonarsource.com/java/RSPEC-1111)

### [Enabling file access for WebViews is security-sensitive](https://rules.sonarsource.com/java/RSPEC-6363)

[Security Hotspot](https://rules.sonarsource.com/java/RSPEC-6363)

### [Enabling JavaScript support for WebViews is security-sensitive](https://rules.sonarsource.com/java/RSPEC-6362)

[Security Hotspot](https://rules.sonarsource.com/java/RSPEC-6362)

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

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

### [Using unencrypted files in mobile applications is security-sensitive](https://rules.sonarsource.com/java/RSPEC-6300)

[Security Hotspot](https://rules.sonarsource.com/java/RSPEC-6300)

### [Using biometric authentication without a cryptographic solution is security-sensitive](https://rules.sonarsource.com/java/RSPEC-6293)

[Security Hotspot](https://rules.sonarsource.com/java/RSPEC-6293)

### [Using unencrypted databases in mobile applications is security-sensitive](https://rules.sonarsource.com/java/RSPEC-6291)

[Security Hotspot](https://rules.sonarsource.com/java/RSPEC-6291)

### [Authorizing non-authenticated users to use keys in the Android KeyStore is security-sensitive](https://rules.sonarsource.com/java/RSPEC-6288)

[Security Hotspot](https://rules.sonarsource.com/java/RSPEC-6288)

### [Allowing user enumeration is security-sensitive](https://rules.sonarsource.com/java/RSPEC-5804)

[Security Hotspot](https://rules.sonarsource.com/java/RSPEC-5804)

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

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

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

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

### [Allowing deserialization of LDAP objects is security-sensitive](https://rules.sonarsource.com/java/RSPEC-4434)

[Security Hotspot](https://rules.sonarsource.com/java/RSPEC-4434)

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

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

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

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

### [Deprecated annotations should include explanations](https://rules.sonarsource.com/java/RSPEC-6355)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-6355)

### [Restricted Identifiers should not be used as Identifiers](https://rules.sonarsource.com/java/RSPEC-6213)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-6213)

### [Redundant constructors/methods should be avoided in records](https://rules.sonarsource.com/java/RSPEC-6207)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-6207)

### [Records should be used instead of ordinary classes when representing immutable data structure](https://rules.sonarsource.com/java/RSPEC-6206)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-6206)

### ["Stream.toList()" method should be used instead of "collectors" when unmodifiable list needed](https://rules.sonarsource.com/java/RSPEC-6204)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-6204)

### [Operator "instanceof" should be used instead of "A.class.isInstance()"](https://rules.sonarsource.com/java/RSPEC-6202)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-6202)

### [String multiline concatenation should be replaced with Text Blocks](https://rules.sonarsource.com/java/RSPEC-6126)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-6126)

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

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

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

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

### [Constructors of an "abstract" class should not be declared "public"](https://rules.sonarsource.com/java/RSPEC-5993)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5993)

### [Similar tests should be grouped in a single Parameterized test](https://rules.sonarsource.com/java/RSPEC-5976)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5976)

### [Tests should be stable](https://rules.sonarsource.com/java/RSPEC-5973)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5973)

### [Test methods should not contain too many assertions](https://rules.sonarsource.com/java/RSPEC-5961)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5961)

### [AssertJ "assertThatThrownBy" should not be used alone](https://rules.sonarsource.com/java/RSPEC-5958)

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

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

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

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

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

### [Regexes containing characters subject to normalization should use the CANON\_EQ flag](https://rules.sonarsource.com/java/RSPEC-5854)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5854)

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

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

### [JUnit assertTrue/assertFalse should be simplified to the corresponding dedicated assertion](https://rules.sonarsource.com/java/RSPEC-5785)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5785)

### [Only one method invocation is expected when testing runtime exceptions](https://rules.sonarsource.com/java/RSPEC-5778)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5778)

### [Exception testing via JUnit ExpectedException rule should not be mixed with other assertions](https://rules.sonarsource.com/java/RSPEC-5776)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5776)

### ["@Deprecated" code marked for removal should never be used](https://rules.sonarsource.com/java/RSPEC-5738)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5738)

### [Vararg method arguments should not be confusing](https://rules.sonarsource.com/java/RSPEC-5669)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5669)

### [Whitespace for text block indent should be consistent](https://rules.sonarsource.com/java/RSPEC-5664)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5664)

### ['List.remove()' should not be used in ascending 'for' loops](https://rules.sonarsource.com/java/RSPEC-5413)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5413)

### [Collection constructors should not be used as java.util.function.Function](https://rules.sonarsource.com/java/RSPEC-5329)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5329)

### ["else" statements should be clearly matched with an "if"](https://rules.sonarsource.com/java/RSPEC-5261)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5261)

### ["Class.forName()" should not load JDBC 4.0+ drivers](https://rules.sonarsource.com/java/RSPEC-4925)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4925)

### [Java features should be preferred to Guava](https://rules.sonarsource.com/java/RSPEC-4738)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4738)

### [Nullness of parameters should be guaranteed](https://rules.sonarsource.com/java/RSPEC-4449)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4449)

### ["Integer.toHexString" should not be used to build hexadecimal strings](https://rules.sonarsource.com/java/RSPEC-4425)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4425)

### [Asserts should not be used to check the parameters of a public method](https://rules.sonarsource.com/java/RSPEC-4274)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4274)

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

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

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

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

### ["java.nio.Files#delete" should be preferred](https://rules.sonarsource.com/java/RSPEC-4042)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4042)

### [Unused "private" classes should be removed](https://rules.sonarsource.com/java/RSPEC-3985)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3985)

### ["Stream.peek" should be used with caution](https://rules.sonarsource.com/java/RSPEC-3864)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3864)

### ["Map.get" and value test should be replaced with single method call](https://rules.sonarsource.com/java/RSPEC-3824)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3824)

### ["@RequestMapping" methods should not be "private"](https://rules.sonarsource.com/java/RSPEC-3751)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3751)

### [Raw types should not be used](https://rules.sonarsource.com/java/RSPEC-3740)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3740)

### ["Arrays.stream" should be used for primitive arrays](https://rules.sonarsource.com/java/RSPEC-3631)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3631)

### [Printf-style format strings should be used correctly](https://rules.sonarsource.com/java/RSPEC-3457)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3457)

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

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

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

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

### ["writeObject" should not be the only "synchronized" code in a class](https://rules.sonarsource.com/java/RSPEC-3042)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3042)

### [Reflection should not be used to increase accessibility of classes, methods, or fields](https://rules.sonarsource.com/java/RSPEC-3011)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3011)

### [Static fields should not be updated in constructors](https://rules.sonarsource.com/java/RSPEC-3010)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3010)

### ["Thread.sleep" should not be used in tests](https://rules.sonarsource.com/java/RSPEC-2925)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2925)

### ["entrySet()" should be iterated when both the key and value are needed](https://rules.sonarsource.com/java/RSPEC-2864)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2864)

### ["DateUtils.truncate" from Apache Commons Lang library should not be used](https://rules.sonarsource.com/java/RSPEC-2718)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2718)

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

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

### ["readObject" should not be "synchronized"](https://rules.sonarsource.com/java/RSPEC-2675)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2675)

### ["Preconditions" and logging arguments should not require evaluation](https://rules.sonarsource.com/java/RSPEC-2629)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2629)

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

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

### ["Lock" objects should not be "synchronized"](https://rules.sonarsource.com/java/RSPEC-2442)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2442)

### [Classes with only "static" methods should not be instantiated](https://rules.sonarsource.com/java/RSPEC-2440)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2440)

### ["Threads" should not be used where "Runnables" are expected](https://rules.sonarsource.com/java/RSPEC-2438)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2438)

### [Inner class calls to super class methods should be unambiguous](https://rules.sonarsource.com/java/RSPEC-2388)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2388)

### [Unused type parameters should be removed](https://rules.sonarsource.com/java/RSPEC-2326)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2326)

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

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

### ["ResultSet.isLast()" should not be used](https://rules.sonarsource.com/java/RSPEC-2232)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2232)

### ["static" members should be accessed statically](https://rules.sonarsource.com/java/RSPEC-2209)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2209)

### [Silly math should not be performed](https://rules.sonarsource.com/java/RSPEC-2185)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2185)

### [Classes named like "Exception" should extend "Exception" or a subclass](https://rules.sonarsource.com/java/RSPEC-2166)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2166)

### [Exceptions should be either logged or rethrown but not both](https://rules.sonarsource.com/java/RSPEC-2139)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2139)

### [Objects should not be created only to "getClass"](https://rules.sonarsource.com/java/RSPEC-2133)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2133)

### [Primitives should not be boxed just for "String" conversion](https://rules.sonarsource.com/java/RSPEC-2131)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2131)

### [Constructors should not be used to instantiate "String", "BigInteger", "BigDecimal" and primitive-wrapper classes](https://rules.sonarsource.com/java/RSPEC-2129)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2129)

### ["URL.hashCode" and "URL.equals" should be avoided](https://rules.sonarsource.com/java/RSPEC-2112)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2112)

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

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

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

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

### ["Object.wait(...)" should never be called on objects that implement "java.util.concurrent.locks.Condition"](https://rules.sonarsource.com/java/RSPEC-1844)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1844)

### [A field should not duplicate the name of its containing class](https://rules.sonarsource.com/java/RSPEC-1700)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1700)

### [JUnit4 @Ignored and JUnit5 @Disabled annotations should be used to disable tests and should provide a rationale](https://rules.sonarsource.com/java/RSPEC-1607)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1607)

### [Anonymous inner classes containing only one method should become lambdas](https://rules.sonarsource.com/java/RSPEC-1604)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1604)

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

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

### ["for" loop stop conditions should be invariant](https://rules.sonarsource.com/java/RSPEC-127)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-127)

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

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

### [Non-constructor methods should not have the same name as the enclosing class](https://rules.sonarsource.com/java/RSPEC-1223)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1223)

### [Exception types should not be tested using "instanceof" in catch blocks](https://rules.sonarsource.com/java/RSPEC-1193)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1193)

### [Classes from "sun.\*" packages should not be used](https://rules.sonarsource.com/java/RSPEC-1191)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1191)

### [Throwable and Error should not be caught](https://rules.sonarsource.com/java/RSPEC-1181)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1181)

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

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

### [Only static class initializers should be used](https://rules.sonarsource.com/java/RSPEC-1171)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1171)

### [Empty arrays and collections should be returned instead of null](https://rules.sonarsource.com/java/RSPEC-1168)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1168)

### ["@Override" should be used on overriding and implementing methods](https://rules.sonarsource.com/java/RSPEC-1161)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1161)

### [Enumeration should not be implemented](https://rules.sonarsource.com/java/RSPEC-1150)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1150)

### [Synchronized classes Vector, Hashtable, Stack and StringBuffer should not be used](https://rules.sonarsource.com/java/RSPEC-1149)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1149)

### [Unused "private" methods should be removed](https://rules.sonarsource.com/java/RSPEC-1144)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1144)

### [Try-catch blocks should not be nested](https://rules.sonarsource.com/java/RSPEC-1141)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1141)

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

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

### [Deprecated elements should have both the annotation and the Javadoc tag](https://rules.sonarsource.com/java/RSPEC-1123)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1123)

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

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

### [Generic exceptions should never be thrown](https://rules.sonarsource.com/java/RSPEC-112)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-112)

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

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

### [Utility classes should not have public constructors](https://rules.sonarsource.com/java/RSPEC-1118)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1118)

### [Local variables should not shadow class fields](https://rules.sonarsource.com/java/RSPEC-1117)

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

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

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

### [Inheritance tree of classes should not be too deep](https://rules.sonarsource.com/java/RSPEC-110)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-110)

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

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

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

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

### [Unused "private" fields should be removed](https://rules.sonarsource.com/java/RSPEC-1068)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1068)

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

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

### [Unused labels should be removed](https://rules.sonarsource.com/java/RSPEC-1065)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1065)

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

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

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

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

### ["ActiveMQConnectionFactory" should not be vulnerable to malicious code deserialization](https://rules.sonarsource.com/java/RSPEC-5301)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-5301)

### [Logging should not be vulnerable to injection attacks](https://rules.sonarsource.com/java/RSPEC-5145)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-5145)

### [Exceptions should not be thrown from servlet methods](https://rules.sonarsource.com/java/RSPEC-1989)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-1989)

### [Return values should not be ignored when they contain the operation status code](https://rules.sonarsource.com/java/RSPEC-899)

[Bug](https://rules.sonarsource.com/java/RSPEC-899)

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

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

### [AssertJ assertions "allMatch" and "doesNotContains" should also test for emptiness](https://rules.sonarsource.com/java/RSPEC-5841)

[Bug](https://rules.sonarsource.com/java/RSPEC-5841)

### [Double Brace Initialization should not be used](https://rules.sonarsource.com/java/RSPEC-3599)

[Bug](https://rules.sonarsource.com/java/RSPEC-3599)

### [Non-primitive fields should not be "volatile"](https://rules.sonarsource.com/java/RSPEC-3077)

[Bug](https://rules.sonarsource.com/java/RSPEC-3077)

### ["toArray" should be passed an array of the proper type](https://rules.sonarsource.com/java/RSPEC-3020)

[Bug](https://rules.sonarsource.com/java/RSPEC-3020)

### [Neither "Math.abs" nor negation should be used on numbers that could be "MIN\_VALUE"](https://rules.sonarsource.com/java/RSPEC-2676)

[Bug](https://rules.sonarsource.com/java/RSPEC-2676)

### [The value returned from a stream read should be checked](https://rules.sonarsource.com/java/RSPEC-2674)

[Bug](https://rules.sonarsource.com/java/RSPEC-2674)

### ["@NonNull" values should not be set to null](https://rules.sonarsource.com/java/RSPEC-2637)

[Bug](https://rules.sonarsource.com/java/RSPEC-2637)

### ["Iterator.next()" methods should throw "NoSuchElementException"](https://rules.sonarsource.com/java/RSPEC-2272)

[Bug](https://rules.sonarsource.com/java/RSPEC-2272)

### ["compareTo" results should not be checked for specific values](https://rules.sonarsource.com/java/RSPEC-2200)

[Bug](https://rules.sonarsource.com/java/RSPEC-2200)

### [Math operands should be cast before assignment](https://rules.sonarsource.com/java/RSPEC-2184)

[Bug](https://rules.sonarsource.com/java/RSPEC-2184)

### [Ints and longs should not be shifted by zero or more than their number of bits-1](https://rules.sonarsource.com/java/RSPEC-2183)

[Bug](https://rules.sonarsource.com/java/RSPEC-2183)

### ["compareTo" should not return "Integer.MIN\_VALUE"](https://rules.sonarsource.com/java/RSPEC-2167)

[Bug](https://rules.sonarsource.com/java/RSPEC-2167)

### [Boxing and unboxing should not be immediately reversed](https://rules.sonarsource.com/java/RSPEC-2153)

[Bug](https://rules.sonarsource.com/java/RSPEC-2153)

### ["equals(Object obj)" should test argument type](https://rules.sonarsource.com/java/RSPEC-2097)

[Bug](https://rules.sonarsource.com/java/RSPEC-2097)

### ["Serializable" inner classes of non-serializable classes should be "static"](https://rules.sonarsource.com/java/RSPEC-2066)

[Bug](https://rules.sonarsource.com/java/RSPEC-2066)

### [The non-serializable super class of a "Serializable" class should have a no-argument constructor](https://rules.sonarsource.com/java/RSPEC-2055)

[Bug](https://rules.sonarsource.com/java/RSPEC-2055)

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

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

### ["equals(Object obj)" and "hashCode()" should be overridden in pairs](https://rules.sonarsource.com/java/RSPEC-1206)

[Bug](https://rules.sonarsource.com/java/RSPEC-1206)

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

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

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

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

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

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

### [Searching OS commands in PATH is security-sensitive](https://rules.sonarsource.com/java/RSPEC-4036)

[Security Hotspot](https://rules.sonarsource.com/java/RSPEC-4036)

### [Allowing both safe and unsafe HTTP methods is security-sensitive](https://rules.sonarsource.com/java/RSPEC-3752)

[Security Hotspot](https://rules.sonarsource.com/java/RSPEC-3752)

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

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

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

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

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

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

### ['serialVersionUID' field should not be set to '0L' in records](https://rules.sonarsource.com/java/RSPEC-6219)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-6219)

### [Permitted types of a sealed class should be omitted if they are declared in the same file](https://rules.sonarsource.com/java/RSPEC-6217)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-6217)

### [Switch arrow labels should not use redundant keywords](https://rules.sonarsource.com/java/RSPEC-6205)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-6205)

### [Text blocks should not be used in complex expressions](https://rules.sonarsource.com/java/RSPEC-6203)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-6203)

### [Pattern Matching for "instanceof" operator should be used instead of simple "instanceof" + cast](https://rules.sonarsource.com/java/RSPEC-6201)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-6201)

### [Call to Mockito method "verify", "when" or "given" should be simplified](https://rules.sonarsource.com/java/RSPEC-6068)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-6068)

### [Character classes should be preferred over reluctant quantifiers in regular expressions](https://rules.sonarsource.com/java/RSPEC-5857)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5857)

### [Consecutive AssertJ "assertThat" statements should be chained](https://rules.sonarsource.com/java/RSPEC-5853)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5853)

### [Chained AssertJ assertions should be simplified to the corresponding dedicated assertion](https://rules.sonarsource.com/java/RSPEC-5838)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5838)

### [Exception testing via JUnit @Test annotation should be avoided](https://rules.sonarsource.com/java/RSPEC-5777)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5777)

### [Escape sequences should not be used in text blocks](https://rules.sonarsource.com/java/RSPEC-5665)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5665)

### [Simple string literal should be used for single line strings](https://rules.sonarsource.com/java/RSPEC-5663)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5663)

### [Boxed "Boolean" should be avoided in boolean expressions](https://rules.sonarsource.com/java/RSPEC-5411)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5411)

### [Type parameters should not shadow other type parameters](https://rules.sonarsource.com/java/RSPEC-4977)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4977)

### ["read(byte[],int,int)" should be overridden](https://rules.sonarsource.com/java/RSPEC-4929)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4929)

### [An iteration on a Collection should be performed on the type handled by the Collection](https://rules.sonarsource.com/java/RSPEC-4838)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4838)

### ["StandardCharsets" constants should be preferred](https://rules.sonarsource.com/java/RSPEC-4719)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4719)

### ["@CheckForNull" or "@Nullable" should not be used on primitive types](https://rules.sonarsource.com/java/RSPEC-4682)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4682)

### [Composed "@RequestMapping" variants should be preferred](https://rules.sonarsource.com/java/RSPEC-4488)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4488)

### ["write(byte[],int,int)" should be overridden](https://rules.sonarsource.com/java/RSPEC-4349)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4349)

### [Functional Interfaces should be as specialised as possible](https://rules.sonarsource.com/java/RSPEC-4276)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4276)

### [Null checks should not be used with "instanceof"](https://rules.sonarsource.com/java/RSPEC-4201)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4201)

### ["close()" calls should not be redundant](https://rules.sonarsource.com/java/RSPEC-4087)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4087)

### ["ThreadLocal.withInitial" should be preferred](https://rules.sonarsource.com/java/RSPEC-4065)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4065)

### ["Stream" call chains should be simplified when possible](https://rules.sonarsource.com/java/RSPEC-4034)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4034)

### [Packages containing only "package-info.java" should be removed](https://rules.sonarsource.com/java/RSPEC-4032)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4032)

### [Arrays should not be created for varargs parameters](https://rules.sonarsource.com/java/RSPEC-3878)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3878)

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

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

### [Test classes should comply with a naming convention](https://rules.sonarsource.com/java/RSPEC-3577)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3577)

### [Loggers should be named for their enclosing classes](https://rules.sonarsource.com/java/RSPEC-3416)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3416)

### [Methods should not return constants](https://rules.sonarsource.com/java/RSPEC-3400)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3400)

### ["private" methods called only by inner classes should be moved to those classes](https://rules.sonarsource.com/java/RSPEC-3398)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3398)

### ["enum" fields should not be publicly mutable](https://rules.sonarsource.com/java/RSPEC-3066)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3066)

### [Abstract methods should not be redundant](https://rules.sonarsource.com/java/RSPEC-3038)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3038)

### [Arrays should not be copied using loops](https://rules.sonarsource.com/java/RSPEC-3012)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3012)

### [Static non-final field names should comply with a naming convention](https://rules.sonarsource.com/java/RSPEC-3008)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3008)

### [JUnit rules should be used](https://rules.sonarsource.com/java/RSPEC-2924)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2924)

### [Nested "enum"s should not be declared static](https://rules.sonarsource.com/java/RSPEC-2786)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2786)

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

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

### [Mutable fields should not be "public static"](https://rules.sonarsource.com/java/RSPEC-2386)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2386)

### [The diamond operator ("<>") should be used](https://rules.sonarsource.com/java/RSPEC-2293)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2293)

### ["finalize" should not set fields to "null"](https://rules.sonarsource.com/java/RSPEC-2165)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2165)

### [Subclasses that add fields should override "equals"](https://rules.sonarsource.com/java/RSPEC-2160)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2160)

### [Catches should be combined](https://rules.sonarsource.com/java/RSPEC-2147)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2147)

### [Methods of "Random" that return floating point values should not be used in random integer generation](https://rules.sonarsource.com/java/RSPEC-2140)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2140)

### [Parsing should be used to convert "Strings" to primitives](https://rules.sonarsource.com/java/RSPEC-2130)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2130)

### [Classes should not be empty](https://rules.sonarsource.com/java/RSPEC-2094)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2094)

### [Fields in non-serializable classes should not be "transient"](https://rules.sonarsource.com/java/RSPEC-2065)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2065)

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

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

### [Redundant casts should not be used](https://rules.sonarsource.com/java/RSPEC-1905)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1905)

### ["@Deprecated" code should not be used](https://rules.sonarsource.com/java/RSPEC-1874)

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

### ["toString()" should never be called on a String object](https://rules.sonarsource.com/java/RSPEC-1858)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1858)

### [Annotation repetitions should not be wrapped](https://rules.sonarsource.com/java/RSPEC-1710)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1710)

### [Multiple variables should not be declared on the same line](https://rules.sonarsource.com/java/RSPEC-1659)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1659)

### [Strings should not be concatenated using '+' in a loop](https://rules.sonarsource.com/java/RSPEC-1643)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1643)

### [Maps with keys that are enum values should be replaced with EnumMap](https://rules.sonarsource.com/java/RSPEC-1640)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1640)

### [Lambdas should be replaced with method references](https://rules.sonarsource.com/java/RSPEC-1612)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1612)

### [Parentheses should be removed from a single lambda input parameter when its type is inferred](https://rules.sonarsource.com/java/RSPEC-1611)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1611)

### [Abstract classes without fields should be converted to interfaces](https://rules.sonarsource.com/java/RSPEC-1610)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1610)

### [Lambdas containing only one statement should not nest this statement in a block](https://rules.sonarsource.com/java/RSPEC-1602)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1602)

### ["Collections.EMPTY\_LIST", "EMPTY\_MAP", and "EMPTY\_SET" should not be used](https://rules.sonarsource.com/java/RSPEC-1596)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1596)

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

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

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

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

### [Private fields only used as local variables in methods should become local variables](https://rules.sonarsource.com/java/RSPEC-1450)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1450)

### ["public static" fields should be constant](https://rules.sonarsource.com/java/RSPEC-1444)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1444)

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

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

### [Declarations should use Java collection interfaces such as "List" rather than specific implementation classes such as "LinkedList"](https://rules.sonarsource.com/java/RSPEC-1319)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1319)

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

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

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

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

### [The default unnamed package should not be used](https://rules.sonarsource.com/java/RSPEC-1220)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1220)

### ["equals(Object obj)" should be overridden along with the "compareTo(T obj)" method](https://rules.sonarsource.com/java/RSPEC-1210)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1210)

### [Package names should comply with a naming convention](https://rules.sonarsource.com/java/RSPEC-120)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-120)

### [Nested code blocks should not be used](https://rules.sonarsource.com/java/RSPEC-1199)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1199)

### [Array designators "[]" should be on the type, not the variable](https://rules.sonarsource.com/java/RSPEC-1197)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1197)

### [Array designators "[]" should be located after the type in method signatures](https://rules.sonarsource.com/java/RSPEC-1195)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1195)

### [Type parameter names should comply with a naming convention](https://rules.sonarsource.com/java/RSPEC-119)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-119)

### [Overriding methods should do more than simply call the same method in the super class](https://rules.sonarsource.com/java/RSPEC-1185)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1185)

### [Classes that override "clone" should be "Cloneable" and call "super.clone()"](https://rules.sonarsource.com/java/RSPEC-1182)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1182)

### [Public constants and fields initialized at declaration should be "static final" rather than merely "final"](https://rules.sonarsource.com/java/RSPEC-1170)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1170)

### [Local variable and method parameter names should comply with a naming convention](https://rules.sonarsource.com/java/RSPEC-117)

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

### [Exception classes should be immutable](https://rules.sonarsource.com/java/RSPEC-1165)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1165)

### [Field names should comply with a naming convention](https://rules.sonarsource.com/java/RSPEC-116)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-116)

### [Primitive wrappers should not be instantiated only for "toString" or "compareTo" calls](https://rules.sonarsource.com/java/RSPEC-1158)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1158)

### [Case insensitive string comparisons should be made without intermediate upper or lower casing](https://rules.sonarsource.com/java/RSPEC-1157)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1157)

### [Collection.isEmpty() should be used to test for emptiness](https://rules.sonarsource.com/java/RSPEC-1155)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1155)

### [String.valueOf() should not be appended to a String](https://rules.sonarsource.com/java/RSPEC-1153)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1153)

### [Interface names should comply with a naming convention](https://rules.sonarsource.com/java/RSPEC-114)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-114)

### ["throws" declarations should not be superfluous](https://rules.sonarsource.com/java/RSPEC-1130)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1130)

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

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

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

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

### [Boolean literals should not be redundant](https://rules.sonarsource.com/java/RSPEC-1125)

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

### [Modifiers should be declared in the correct order](https://rules.sonarsource.com/java/RSPEC-1124)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1124)

### [Empty statements should be removed](https://rules.sonarsource.com/java/RSPEC-1116)

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

### [Class variable fields should not have public accessibility](https://rules.sonarsource.com/java/RSPEC-1104)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1104)

### [URIs should not be hardcoded](https://rules.sonarsource.com/java/RSPEC-1075)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1075)

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

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

### [Method names should comply with a naming convention](https://rules.sonarsource.com/java/RSPEC-100)

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

### [Comma-separated labels should be used in Switch with colon case](https://rules.sonarsource.com/java/RSPEC-6208)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-6208)

### [JUnit5 test classes and methods should have default package visibility](https://rules.sonarsource.com/java/RSPEC-5786)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5786)

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

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

### [Deprecated code should be removed](https://rules.sonarsource.com/java/RSPEC-1133)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1133)

### [Annotated Mockito objects should be initialized](https://rules.sonarsource.com/java/RSPEC-5979)

[Bug](https://rules.sonarsource.com/java/RSPEC-5979)

### [Custom resources should be closed](https://rules.sonarsource.com/java/RSPEC-3546)

[Bug](https://rules.sonarsource.com/java/RSPEC-3546)

### [Threads should not be started in constructors](https://rules.sonarsource.com/java/RSPEC-2693)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2693)

### ["main" should not "throw" anything](https://rules.sonarsource.com/java/RSPEC-2096)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2096)

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

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

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

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

### [Exit methods should not be called](https://rules.sonarsource.com/java/RSPEC-1147)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1147)

### [HTTP response headers should not be vulnerable to injection attacks](https://rules.sonarsource.com/java/RSPEC-5167)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-5167)

### [Members of Spring components should be injected](https://rules.sonarsource.com/java/RSPEC-3749)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-3749)

### [Classes should not be loaded dynamically](https://rules.sonarsource.com/java/RSPEC-2658)

[Vulnerability](https://rules.sonarsource.com/java/RSPEC-2658)

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

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

### ["Bean Validation" (JSR 380) should be properly configured](https://rules.sonarsource.com/java/RSPEC-5128)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5128)

### [Spring beans should be considered by "@ComponentScan"](https://rules.sonarsource.com/java/RSPEC-4605)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4605)

### [Number patterns should be regular](https://rules.sonarsource.com/java/RSPEC-3937)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3937)

### [Lazy initialization of "static" fields should be "synchronized"](https://rules.sonarsource.com/java/RSPEC-2444)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2444)

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

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

### [Modulus results should not be checked for direct equality](https://rules.sonarsource.com/java/RSPEC-2197)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2197)

### [Comparators should be "Serializable"](https://rules.sonarsource.com/java/RSPEC-2063)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2063)

### ["Serializable" classes should have a "serialVersionUID"](https://rules.sonarsource.com/java/RSPEC-2057)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2057)

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

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

### [Constructors should only call non-overridable methods](https://rules.sonarsource.com/java/RSPEC-1699)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1699)

### [Methods should not be too complex](https://rules.sonarsource.com/java/RSPEC-1541)

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

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

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

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

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

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

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

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

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

### [Mockito argument matchers should be used on all parameters](https://rules.sonarsource.com/java/RSPEC-6073)

[Bug](https://rules.sonarsource.com/java/RSPEC-6073)

### [Spring "@Controller" classes should not use "@Scope"](https://rules.sonarsource.com/java/RSPEC-3750)

[Bug](https://rules.sonarsource.com/java/RSPEC-3750)

### [Constructor injection should be used instead of field injection](https://rules.sonarsource.com/java/RSPEC-3306)

[Bug](https://rules.sonarsource.com/java/RSPEC-3306)

### [Classes that don't define "hashCode()" should not be used in hashes](https://rules.sonarsource.com/java/RSPEC-2141)

[Bug](https://rules.sonarsource.com/java/RSPEC-2141)

### [Floating point numbers should not be tested for equality](https://rules.sonarsource.com/java/RSPEC-1244)

[Bug](https://rules.sonarsource.com/java/RSPEC-1244)

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

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

### [Limited dependence should be placed on operator precedence](https://rules.sonarsource.com/java/RSPEC-864)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-864)

### [Custom getter method should not be used to override record's getter behavior](https://rules.sonarsource.com/java/RSPEC-6211)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-6211)

### [Tests should use fixed data instead of randomized data](https://rules.sonarsource.com/java/RSPEC-5977)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5977)

### [Spring's ModelAndViewAssert assertions should be used instead of other assertions](https://rules.sonarsource.com/java/RSPEC-5970)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5970)

### [Lambdas should not have too many lines](https://rules.sonarsource.com/java/RSPEC-5612)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5612)

### ["@EnableAutoConfiguration" should be fine-tuned](https://rules.sonarsource.com/java/RSPEC-4604)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4604)

### [Enum values should be compared with "=="](https://rules.sonarsource.com/java/RSPEC-4551)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4551)

### [Spring components should use constructor injection](https://rules.sonarsource.com/java/RSPEC-4288)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4288)

### [Regex patterns should not be created needlessly](https://rules.sonarsource.com/java/RSPEC-4248)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4248)

### [Track uses of disallowed constructors](https://rules.sonarsource.com/java/RSPEC-4011)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4011)

### [Java 8's "Files.exists" should not be used](https://rules.sonarsource.com/java/RSPEC-3725)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3725)

### ["Optional" should not be used for parameters](https://rules.sonarsource.com/java/RSPEC-3553)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3553)

### [Tests should be kept in a dedicated source directory](https://rules.sonarsource.com/java/RSPEC-3414)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3414)

### ["this" should not be exposed from constructors](https://rules.sonarsource.com/java/RSPEC-3366)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3366)

### [Classes should not have too many "static" imports](https://rules.sonarsource.com/java/RSPEC-3030)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3030)

### [Escaped Unicode characters should not be used](https://rules.sonarsource.com/java/RSPEC-2973)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2973)

### [Inner classes should not have too many lines of code](https://rules.sonarsource.com/java/RSPEC-2972)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2972)

### [Inner classes which do not reference their owning classes should be "static"](https://rules.sonarsource.com/java/RSPEC-2694)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2694)

### ["deleteOnExit" should not be used](https://rules.sonarsource.com/java/RSPEC-2308)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2308)

### [Public methods should not contain selector arguments](https://rules.sonarsource.com/java/RSPEC-2301)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2301)

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

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

### [Track uses of disallowed methods](https://rules.sonarsource.com/java/RSPEC-2253)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2253)

### [Types should be used in lambdas](https://rules.sonarsource.com/java/RSPEC-2211)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2211)

### ["java.time" classes should be used for dates and times](https://rules.sonarsource.com/java/RSPEC-2143)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2143)

### [The names of methods with boolean return values should start with "is" or "has"](https://rules.sonarsource.com/java/RSPEC-2047)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2047)

### [Files should contain only one top-level class or interface each](https://rules.sonarsource.com/java/RSPEC-1996)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1996)

### [Classes should not have too many fields](https://rules.sonarsource.com/java/RSPEC-1820)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1820)

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

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

### [Standard functional interfaces should not be redefined](https://rules.sonarsource.com/java/RSPEC-1711)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1711)

### ["NullPointerException" should not be caught](https://rules.sonarsource.com/java/RSPEC-1696)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1696)

### ["NullPointerException" should not be explicitly thrown](https://rules.sonarsource.com/java/RSPEC-1695)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1695)

### [Classes should not have too many methods](https://rules.sonarsource.com/java/RSPEC-1448)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1448)

### [Methods should not have too many lines](https://rules.sonarsource.com/java/RSPEC-138)

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

### [Track uses of "NOSONAR" comments](https://rules.sonarsource.com/java/RSPEC-1291)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1291)

### [Classes and enums with private members should have a constructor](https://rules.sonarsource.com/java/RSPEC-1258)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1258)

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

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

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

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

### [Classes should not be coupled to too many other classes (Single Responsibility Principle)](https://rules.sonarsource.com/java/RSPEC-1200)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1200)

### ["java.lang.Error" should not be extended](https://rules.sonarsource.com/java/RSPEC-1194)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1194)

### [Anonymous classes should not have too many lines](https://rules.sonarsource.com/java/RSPEC-1188)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1188)

### [Public types, methods and fields (API) should be documented with Javadoc](https://rules.sonarsource.com/java/RSPEC-1176)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1176)

### [Exception handlers should preserve the original exceptions](https://rules.sonarsource.com/java/RSPEC-1166)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1166)

### [Checked exceptions should not be thrown](https://rules.sonarsource.com/java/RSPEC-1162)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1162)

### [Public methods should throw at most one checked exception](https://rules.sonarsource.com/java/RSPEC-1160)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1160)

### ["switch case" clauses should not have too many lines of code](https://rules.sonarsource.com/java/RSPEC-1151)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1151)

### [Methods should not have too many return statements](https://rules.sonarsource.com/java/RSPEC-1142)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1142)

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

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

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

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

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

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

### [JEE applications should not "getClassLoader"](https://rules.sonarsource.com/java/RSPEC-3032)

[Bug](https://rules.sonarsource.com/java/RSPEC-3032)

### [Math should not be performed on floats](https://rules.sonarsource.com/java/RSPEC-2164)

[Bug](https://rules.sonarsource.com/java/RSPEC-2164)

### ["equals" methods should be symmetric and work for subclasses](https://rules.sonarsource.com/java/RSPEC-2162)

[Bug](https://rules.sonarsource.com/java/RSPEC-2162)

### [Literal suffixes should be upper case](https://rules.sonarsource.com/java/RSPEC-818)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-818)

### [Unicode-aware versions of character classes should be preferred](https://rules.sonarsource.com/java/RSPEC-5867)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5867)

### [Use Java 12 "switch" expression](https://rules.sonarsource.com/java/RSPEC-5194)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5194)

### ["serialVersionUID" should not be declared blindly](https://rules.sonarsource.com/java/RSPEC-4926)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4926)

### ["Stream.collect()" calls should not be redundant](https://rules.sonarsource.com/java/RSPEC-4266)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4266)

### [Local constants should follow naming conventions for constants](https://rules.sonarsource.com/java/RSPEC-4174)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-4174)

### [Unit tests should throw exceptions](https://rules.sonarsource.com/java/RSPEC-3658)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3658)

### [Test methods should comply with a naming convention](https://rules.sonarsource.com/java/RSPEC-3578)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3578)

### [Value-based objects should not be serialized](https://rules.sonarsource.com/java/RSPEC-3437)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3437)

### [Default annotation parameter values should not be passed as arguments](https://rules.sonarsource.com/java/RSPEC-3254)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3254)

### [Method parameters should be declared with base types](https://rules.sonarsource.com/java/RSPEC-3242)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3242)

### [Fields should not be initialized to default values](https://rules.sonarsource.com/java/RSPEC-3052)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3052)

### [Multiple loops over the same set should be combined](https://rules.sonarsource.com/java/RSPEC-3047)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3047)

### [Classes without "public" constructors should be "final"](https://rules.sonarsource.com/java/RSPEC-2974)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2974)

### [Unnecessary semicolons should be omitted](https://rules.sonarsource.com/java/RSPEC-2959)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2959)

### [Literal boolean values and nulls should not be used in assertions](https://rules.sonarsource.com/java/RSPEC-2701)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2701)

### [Test assertions should include messages](https://rules.sonarsource.com/java/RSPEC-2698)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2698)

### [Mutable members should not be stored or returned directly](https://rules.sonarsource.com/java/RSPEC-2384)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2384)

### [Redundant modifiers should not be used](https://rules.sonarsource.com/java/RSPEC-2333)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2333)

### ["private" and "final" methods that don't access instance data should be "static"](https://rules.sonarsource.com/java/RSPEC-2325)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2325)

### [Files should not be empty](https://rules.sonarsource.com/java/RSPEC-2309)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2309)

### [Collection methods with O(n) performance should be used carefully](https://rules.sonarsource.com/java/RSPEC-2250)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2250)

### ["Exception" should not be caught when not required by called methods](https://rules.sonarsource.com/java/RSPEC-2221)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2221)

### ["collect" should be used with "Streams" instead of "list::add"](https://rules.sonarsource.com/java/RSPEC-2203)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2203)

### [Switches should be used for sequences of simple "String" tests](https://rules.sonarsource.com/java/RSPEC-2196)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2196)

### ["final" classes should not have "protected" members](https://rules.sonarsource.com/java/RSPEC-2156)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2156)

### [Underscores should be used to make large numbers readable](https://rules.sonarsource.com/java/RSPEC-2148)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2148)

### ["Serializable" inner classes of "Serializable" classes should be static](https://rules.sonarsource.com/java/RSPEC-2059)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2059)

### [Member variable visibility should be specified](https://rules.sonarsource.com/java/RSPEC-2039)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-2039)

### [Classes and methods that rely on the default system encoding should not be used](https://rules.sonarsource.com/java/RSPEC-1943)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1943)

### [Simple class names should be used](https://rules.sonarsource.com/java/RSPEC-1942)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1942)

### [Variables should not be declared before they are relevant](https://rules.sonarsource.com/java/RSPEC-1941)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1941)

### [Extensions and implementations should not be redundant](https://rules.sonarsource.com/java/RSPEC-1939)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1939)

### ["==" and "!=" should not be used when "equals" is overridden](https://rules.sonarsource.com/java/RSPEC-1698)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1698)

### [An abstract class should have both abstract and concrete methods](https://rules.sonarsource.com/java/RSPEC-1694)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1694)

### [Sets with elements that are enum values should be replaced with EnumSet](https://rules.sonarsource.com/java/RSPEC-1641)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1641)

### [String operations should not rely on the default system locale](https://rules.sonarsource.com/java/RSPEC-1449)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1449)

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

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

### [Track uses of "CHECKSTYLE:OFF" suppression comments](https://rules.sonarsource.com/java/RSPEC-1315)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1315)

### [Loggers should be "private static final" and should share a naming convention](https://rules.sonarsource.com/java/RSPEC-1312)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1312)

### [Track uses of "NOPMD" suppression comments](https://rules.sonarsource.com/java/RSPEC-1310)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1310)

### [Packages should have a javadoc file 'package-info.java'](https://rules.sonarsource.com/java/RSPEC-1228)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1228)

### [The members of an interface or class declaration should appear in a pre-defined order](https://rules.sonarsource.com/java/RSPEC-1213)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1213)

### [Abstract class names should comply with a naming convention](https://rules.sonarsource.com/java/RSPEC-118)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-118)

### [Strings literals should be placed on the left side when checking for equality](https://rules.sonarsource.com/java/RSPEC-1132)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1132)

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

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

### [Source code should be indented consistently](https://rules.sonarsource.com/java/RSPEC-1120)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1120)

### [A close curly brace should be located at the beginning of a line](https://rules.sonarsource.com/java/RSPEC-1109)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1109)

### [Close curly brace and the next "else", "catch" and "finally" keywords should be on two different lines](https://rules.sonarsource.com/java/RSPEC-1108)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1108)

### [Close curly brace and the next "else", "catch" and "finally" keywords should be located on the same line](https://rules.sonarsource.com/java/RSPEC-1107)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1107)

### [An open curly brace should be located at the beginning of a line](https://rules.sonarsource.com/java/RSPEC-1106)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1106)

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

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

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

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

### [Functions should not be defined with a variable number of arguments](https://rules.sonarsource.com/java/RSPEC-923)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-923)

### [Local-Variable Type Inference should be used](https://rules.sonarsource.com/java/RSPEC-6212)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-6212)

### [Migrate your tests from JUnit4 to the new JUnit5 annotations](https://rules.sonarsource.com/java/RSPEC-5793)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-5793)

### [Track uses of disallowed classes](https://rules.sonarsource.com/java/RSPEC-3688)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-3688)

### [Track uses of "@SuppressWarnings" annotations](https://rules.sonarsource.com/java/RSPEC-1309)

[Code Smell](https://rules.sonarsource.com/java/RSPEC-1309)