


 sonar RULES


Products


▼


 Secrets


 ABAP


 Apex


 C


 C++


 CloudFormation


 COBOL


 C#


 CSS


 Flex


 Go


 HTML


 **Java**


 JavaScript


 Kotlin


 Objective C


 PHP


 PL/I


 PL/SQL


 Python


 RPG


 Ruby


 Scala


 Swift


 Terraform


 Text


 TypeScript

 T-SQL

 VB.NET

 VB6

 XML



Java static code analysis

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

All rules632

Vulnerability53

Bug154

Security Hotspot36

Code Smell389

Quick Fix42

Tags

Search by name...

Resources should be closed

Bug

Hard-coded credentials are security-sensitive

Security Hotspot

Methods returns should not be invariant

Code Smell

"ThreadGroup" should not be used

Code Smell

"clone" should not be overridden

Code Smell

Assertions should be complete

Code Smell

Tests should include assertions

Code Smell

Silly bit operations should not be performed

Code Smell

Child class fields should not shadow parent class fields

Code Smell

JUnit test cases should call super methods

Code Smell

TestCases should contain tests

Code Smell

Short-circuit logic should be used in boolean contexts

Code Smell

Methods and field names should not be the same or differ only by

"@Controller" classes that use "@SessionAttributes" must call "setComplete" on their "SessionStatus" objects

Analyze your code

BugBlockerspring

A Spring @Controller that uses @SessionAttributes is designed to handle a stateful / multi-post form. Such @Controllers use the specified @SessionAttributes to store data on the server between requests. That data should be cleaned up when the session is over, but unless setComplete() is called on the SessionStatus object from a @RequestMapping method, neither Spring nor the JVM will know it's time to do that. Note that the SessionStatus object must be passed to that method as a parameter.

Noncompliant Code Example

```
@Controller
@SessionAttributes("hello") // Noncompliant; this doesn't g
public class HelloWorld {

    @RequestMapping("/greet", method = GET)
    public String greet(String greetee) {

        return "Hello " + greetee;
    }
}
```

Compliant Solution

```
@Controller
@SessionAttributes("hello")
public class HelloWorld {

    @RequestMapping("/greet", method = GET)
    public String greet(String greetee) {

        return "Hello " + greetee;
    }




    @RequestMapping("/goodbye", method = POST)
    public String goodbye(SessionStatus status) {
        //...
        status.setComplete();
    }
}
```

Available In:

sonarlint | sonarcloud | sonarqube

https://rules.sonarsource.com/java/RSPEC-3753

1/2

<div>Switch cases should end with an unconditional "break" statement</div> <div> Code Smell</div>
<div>"switch" statements should not contain non-case labels</div> <div> Code Smell</div>
<div>Future keywords should not be used as names</div> <div> Code Smell</div>
<div>Thread suspensions should not be</div>

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.
[Privacy Policy](#)