




 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 rules **632**

Vulnerability **53**


Bug **154**


Security Hotspot **36**

Code Smell **389**


Quick Fix **42**


Tags ▾

Search by name... 


 Code Smell


Abstract class names should comply with a naming convention

 Code Smell


 Code Smell


Strings literals should be placed on the left side when checking for equality

 Code Smell


 Code Smell


Files should contain an empty newline at the end

 Code Smell


 Code Smell


Source code should be indented consistently

 Code Smell


 Code Smell


A close curly brace should be located at the beginning of a line

 Code Smell


 Code Smell


Close curly brace and the next "else", "catch" and "finally" keywords should be on two different lines

 Code Smell

 Code Smell

Close curly brace and the next "else", "catch" and "finally" keywords should be located on the same line

 Code Smell

 Code Smell

An open curly brace should be located at the beginning of a line

 Code Smell

 Code Smell

An open curly brace should be located at the end of a line

 Code Smell

 Code Smell

Tabulation characters should not be used

 Code Smell

 Code Smell

Functions should not be defined with a variable number of arguments





 Code Smell

 Code Smell

Local-Variable Type Inference should be used

 Code Smell

Migrate your tests from JUnit4 to the new JUnit5 annotations

 Code Smell  Info   junit tests

As mentioned in JUnit5 documentation, it is possible to integrate JUnit4 with JUnit5:

JUnit provides a gentle migration path via a JUnit Vintage test engine which allows existing tests based on JUnit 3 and JUnit 4 to be executed using the JUnit Platform infrastructure. Since all classes and annotations specific to JUnit Jupiter reside under a new org.junit.jupiter base package, having both JUnit 4 and JUnit Jupiter in the classpath does not lead to any conflicts.

However, maintaining both systems is a temporary solution. This rule flags all the annotations from JUnit4 which would need to be migrated to JUnit5, hence helping migration of a project.

Here is the list of JUnit4 annotations tracked by the rule, with their corresponding annotations in JUnit5:

JUnit4	JUnit5
org.junit.Test	org.junit.jupiter.api.Test
org.junit.Before	org.junit.jupiter.api.BeforeEach
org.junit.After	org.junit.jupiter.api.AfterEach
org.junit.BeforeClass	org.junit.jupiter.api.BeforeAll
org.junit.AfterClass	org.junit.jupiter.api.AfterAll
org.junit.Ignore	org.junit.jupiter.api.Disabled

Note that the following annotations might requires some rework of the tests to have JUnit5 equivalent behavior. A simple replacement of the annotation won't work immediately:

JUnit4	JUnit5
org.junit.experimental.categories.Category	org.junit.jupiter.api.Tag
org.junit.Rule	org.junit.jupiter.api.extension.ExtensionRule
org.junit.ClassRule	org.junit.jupiter.api.extension.ExtensionRule
org.junit.runner.RunWith	org.junit.jupiter.api.extension.ExtensionRule

Noncompliant Code Example

```
package org.foo;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Ignore;
import org.junit.Test;
import org.junit.experimental.categories.Category;
import org.junit.runner.RunWith;




@RunWith(MyJUnit4Runner.class)
public class MyJUnit4Test {

    @BeforeClass
    public static void beforeAll() {
        System.out.println("beforeAll");
    }

    @AfterClass
    public static void afterAll() {
        System.out.println("AfterAll");
    }
}
```

https://rules.sonarsource.com/java/RSPEC-5793

1/3

<div>Migrate your tests from JUnit4 to the new JUnit5 annotations</div> <div> Code Smell</div>
<div>Track uses of disallowed classes</div> <div> Code Smell</div>
<div>Track uses of "@SuppressWarnings" annotations</div> <div> Code Smell</div>

```
@Before
public void beforeEach() {
    System.out.println("beforeEach");
}

@After
public void afterEach() {
    System.out.println("afterEach");
}

@Test
public void test1() throws Exception {
    System.out.println("test1");
}

public interface SomeTests { /* category marker */ }

@Test
@Category(SomeTests.class)
public void test2() throws Exception {
    System.out.println("test2");
}

@Test
@Ignore("Requires fix of #42")
public void ignored() throws Exception {
    System.out.println("ignored");
}
}
```

Compliant Solution

```
package org.foo;

import org.junit.jupiter.api.AfterAll;
import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Disabled;
import org.junit.jupiter.api.Tag;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.extension.ExtendWith;

@ExtendWith(MyJUnit5Extension.class)
class MyJUnit5Test {

    @BeforeAll
    static void beforeAll() {
        System.out.println("beforeAll");
    }

    @AfterAll
    static void afterAll() {
        System.out.println("afterAll");
    }

    @BeforeEach
    void beforeEach() {
        System.out.println("beforeEach");
    }

    @AfterEach
    void afterEach() {
        System.out.println("afterEach");
    }

    @Test
    void test1() {
        System.out.println("test1");
    }

    @Test
    @Tag("SomeTests")
    void test2() {
        System.out.println("test2");
    }

    @Test
    @Disabled("Requires fix of #42")
    void disabled() {
        System.out.println("ignored");
    }
}
```

See

- [JUnit 5: Migrating from JUnit4](#)

Available In:

sonarlint  | **sonarcloud**  | **sonarqube** 

© 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](#)