




 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... 🔍

Abstract class names should comply with a naming convention

Code Smell

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

Code Smell

Files should contain an empty newline at the end

Code Smell

Source code should be indented consistently

Code Smell

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

Code Smell

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

Code Smell

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

Code Smell

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

Code Smell

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

Code Smell

Tabulation characters should not be used

Code Smell

Functions should not be defined with a variable number of arguments

Code Smell

Annotated Mockito objects should be initialized

Analyze your code

Bug

Blocker

tests mockito

Objects annotated with Mockito annotations `@Mock`, `@Spy`, `@Captor`, or `@InjectMocks` need to be initialized explicitly.

There are several ways to do this:

- Call `MockitoAnnotations.openMocks(this)` or `MockitoAnnotations.initMocks(this)` in a setup method
- Annotate test class with `@RunWith(MockitoJUnitRunner.class)` (JUnit 4)
- Annotate test class with `@ExtendWith(MockitoExtension.class)` (JUnit 5 Jupiter)
- Use `@Rule public MockitoRule rule = MockitoJUnit.rule();`

Test using uninitialized mocks will fail.

Note that this only applies to annotated Mockito objects. It is not necessary to initialize objects instantiated via `Mockito.mock()` or `Mockito.spy()`.

This rule raises an issue when a test class uses uninitialized mocks.

Noncompliant Code Example

```
public class FooTest { // Noncompliant: Mockito initialization
    @Mock private Bar bar;

    @Spy private Baz baz;

    @InjectMocks private Foo fooUnderTest;

    @Test
    void someTest() {
        // test something ...
    }

    @Nested
    public class Nested {
        @Mock
        private Bar bar;
    }
}
```





Compliant Solution

```
@RunWith(MockitoJUnitRunner.class)
public class FooTest {
    @Mock private Bar bar;
    // ...
}
```

```
@ExtendWith(MockitoExtension.class)
public class FooTest {
```

https://rules.sonarsource.com/java/RSPEC-5979

1/2

Local-Variable Type Inference should be used
 Code Smell
Migrate your tests from JUnit4 to the new JUnit5 annotations
 Code Smell
Track uses of disallowed classes
 Code Smell
Track uses of "@SuppressWarnings" annotations
 Code Smell

```
@Mock private Bar bar;
// ...
}
```

```
public class FooTest {
    @Rule
    public MockitoRule rule = MockitoJUnit.rule();

    @Mock private Bar bar;
    // ...
}
```

```
public class FooTest {
    @Mock private Bar bar;
    // ...

    @BeforeEach
    void setUp() {
        MockitoAnnotations.openMocks(this);
    }
    // ...
}
```

```
public class FooTest {
    @Mock private Bar bar;
    // ...

    @Before
    void setUp() {
        MockitoAnnotations.initMocks(this);
    }
    // ...
}
```

```
@ExtendWith(MockitoExtension.class)
public class FooTest {
    @Nested
    public class Nested {
        @Mock
        private Bar bar;
    }
}
```

See

- [Mockito documentation - MockitoAnnotations](#)
- [Mockito documentation - MockitoRule](#)

Available In:

