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 rules 632 | Vulnerability 53 | Bug 154 | Security Hotspot 36 | Code Smell 389 | Quick Fix 42

Tags | Search by name...

Security Hotspot

Using biometric authentication without a cryptographic solution is security-sensitive
Security Hotspot

Using unencrypted databases in mobile applications is security-sensitive
Security Hotspot

Authorizing non-authenticated users to use keys in the Android KeyStore is security-sensitive
Security Hotspot

Allowing user enumeration is security-sensitive
Security Hotspot

Allowing requests with excessive content length is security-sensitive
Security Hotspot

Disabling auto-escaping in template engines is security-sensitive
Security Hotspot

Allowing deserialization of LDAP objects is security-sensitive
Security Hotspot

Setting loose POSIX file permissions is security-sensitive
Security Hotspot

Formatting SQL queries is security-sensitive
Security Hotspot

Deprecated annotations should include explanations
Code Smell

Restricted Identifiers should not be used as Identifiers

Methods should not be empty

Analyze your code

Code Smell   Critical   Quick Fix   suspicious

There are several reasons for a method not to have a method body:

- It is an unintentional omission, and should be fixed to prevent an unexpected behavior in production.
- It is not yet, or never will be, supported. In this case an `UnsupportedOperationException` should be thrown.
- The method is an intentionally-blank override. In this case a nested comment should explain the reason for the blank override.

Noncompliant Code Example

```
public void doSomething() {
}

public void doSomethingElse() {
}
```

Compliant Solution

```
@Override
public void doSomething() {
  // Do nothing because of X and Y.
}

@Override
public void doSomethingElse() {
  throw new UnsupportedOperationException();
}
```

Exceptions

Default (no-argument) constructors are ignored when there are other constructors in the class, as are empty methods in abstract classes.

```
public abstract class Animal {
  void speak() {  // default implementation ignored
  }
}
```

Available In:

sonarlint | sonarcloud | sonarqube

Code Smell

Redundant constructors/methods should be avoided in records

Code Smell

Records should be used instead of ordinary classes when representing immutable data structure

Code Smell

"Stream.toList()" method should be used instead of "collectors" when unmodifiable list needed

Code Smell