




 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... 🔍

 Code Smell


Boxed "Boolean" should be avoided in boolean expressions




Type parameters should not shadow other type parameters




"read(byte[],int,int)" should be overridden




An iteration on a Collection should be performed on the type handled by the Collection




"StandardCharsets" constants should be preferred




"@CheckForNull" or "@Nullable" should not be used on primitive types




Composed "@RequestMapping" variants should be preferred




"write(byte[],int,int)" should be overridden




Functional Interfaces should be as specialised as possible



Null checks should not be used with "instanceof"






"close()" calls should not be redundant



Custom serialization method signatures should meet requirements

Analyze your code

 Bug  Major  cert

Writers of `Serializable` classes can choose to let Java's automatic mechanisms handle serialization and deserialization, or they can choose to handle it themselves by implementing specific methods. However, if the signatures of those methods are not exactly what is expected, they will be ignored and the default serialization mechanisms will kick back in.

Noncompliant Code Example

```
public class Watermelon implements Serializable {
    // ...
    void writeObject(java.io.ObjectOutputStream out) // Noncompliant
        throws IOException
    {...}

    private void readObject(java.io.ObjectInputStream in)
    {...}

    public void readObjectNoData() // Noncompliant; not private
    {...}

    static Object readResolve() throws ObjectStreamException

    Watermelon writeReplace() throws ObjectStreamException // Noncompliant
    {...}
}
```

Compliant Solution

```
public class Watermelon implements Serializable {
    // ...
    private void writeObject(java.io.ObjectOutputStream out)
        throws IOException
    {...}

    private void readObject(java.io.ObjectInputStream in)
        throws IOException, ClassNotFoundException
    {...}





    private void readObjectNoData()
        throws ObjectStreamException
    {...}

    protected Object readResolve() throws ObjectStreamException
    {...}

    private Object writeReplace() throws ObjectStreamException
    {...}
}
```

https://rules.sonarsource.com/java/RSPEC-2061

1/2

<div><div>"ThreadLocal.withInitial" should be preferred</div><div> Code Smell</div></div>	<div><div>See</div><div><ul style="list-style-type: none"><li><a href="#">CERT, SER01-J</a>. - Do not deviate from the proper signatures of serialization methods</li></ul></div><div><div>Available In:</div><div><div>sonarlint</div><div>sonarcloud</div><div>sonarqube</div></div></div></div>
<div><div>"Stream" call chains should be simplified when possible</div><div> Code Smell</div></div>	
<div><div>Packages containing only "package-info.java" should be removed</div><div> Code Smell</div></div>	
<div><div>Arrays should not be created for varargs parameters</div><div> Code Smell</div></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](#)