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...

Guava

Code Smell

Nullness of parameters should be guaranteed

Code Smell

"Integer.toHexString" should not be used to build hexadecimal strings

Code Smell

Asserts should not be used to check the parameters of a public method

Code Smell

Assignments should not be redundant

Code Smell

Methods should not have identical implementations

Code Smell

"java.nio.Files#delete" should be preferred

Code Smell

Unused "private" classes should be removed

Code Smell

"Stream.peek" should be used with caution

Code Smell

"Map.get" and value test should be replaced with single method call

Code Smell

"@RequestMapping" methods should not be "private"

Code Smell

Raw types should not be used

Code Smell

DateTimeFormatters should not use mismatched year and week numbers

Analyze your code

Bug Major

When creating a `DateTimeFormatter` using the `WeekFields.weekBasedYear()` temporal field, the resulting year number may be off by 1 at the beginning of a new year (when the date to format is in a week that is shared by two consecutive years).

Using this year number in combination with an incompatible week temporal field yields a result that may be confused with the first week of the previous year.

Instead, when paired with a week temporal field, the week-based year should only be used with the week of week-based year temporal field `WeekFields.weekOfWeekBasedYear()`.

Alternatively the temporal field `ChronoField.ALIGNED_WEEK_OF_YEAR` can be used together with a regular year (but not the week based year).

Noncompliant Code Example

```
new DateTimeFormatterBuilder()
      .appendValue(ChronoField.YEAR, 4) // Noncompliant: usi
      .appendLiteral('-')
      .appendValue(WeekFields.ISO.weekOfWeekBasedYear(), 2)
      .toFormatter();

new DateTimeFormatterBuilder()
      .appendValue(ChronoField.YEAR_OF_ERA, 4) // Noncomplia
      .appendLiteral('-')
      .appendValue(WeekFields.ISO.weekOfWeekBasedYear(), 2)
      .toFormatter();

new DateTimeFormatterBuilder()
      .appendValue(WeekFields.ISO.weekBasedYear(), 4) // Non
      .appendLiteral('-')
      .appendValue(ChronoField.ALIGNED_WEEK_OF_YEAR, 2)
      .toFormatter();
```

Here the first two formatters would wrongly format the 1st of January 2016 as "2016-53" while the last one would format it as "2015-01"

Compliant Solution

```
new DateTimeFormatterBuilder()
      .appendValue(WeekFields.ISO.weekBasedYear(), 4)
      .appendLiteral('-')
      .appendValue(WeekFields.ISO.weekOfWeekBasedYear(), 2)
      .toFormatter();

new DateTimeFormatterBuilder()
      .appendValue(ChronoField.YEAR, 4)
      .appendLiteral('-')
      .appendValue(ChronoField.ALIGNED_WEEK_OF_YEAR, 2)
```

```
        .toFormatter();

new DateTimeFormatterBuilder()
        .appendValue(ChronoField.YEAR_OF_ERA, 4)
        .appendLiteral('-')
        .appendValue(ChronoField.ALIGNED_WEEK_OF_YEAR, 2)
        .toFormatter();
```

Here the first formatter would format the 1st of January 2016 as "2015-53" while the last two would produce "2016-01", both of which are correct depending on how you count the weeks.

Exceptions

No issue is raised when week-based year is not used in combination with a week temporal field.

Similarly, no issue is raised if week of week-based year is not used in combination with a year temporal field.

Available In:

sonarlint | sonarcloud | sonarqube

"Arrays.stream" should be used for primitive arrays

Code Smell

Printf-style format strings should be used correctly

Code Smell

Assertion arguments should be passed in the correct order

Code Smell

Ternary operators should not be nested

Code Smell

Privacy Policy