

sonarqube



Code inspection tool

A presentation by Kevin Reis | 11/14/2019 | TINF18B4

About SonarQube



“SonarQube empowers all developers to write cleaner and safer code.”

~ <https://www.sonarqube.org/>

- **Code inspection tool**
- **Can be used in CI/CD environment**
- **Available with probably many build management tools**

Multi-language Support



TypeScript



Python



Swift

COBOL

Apex

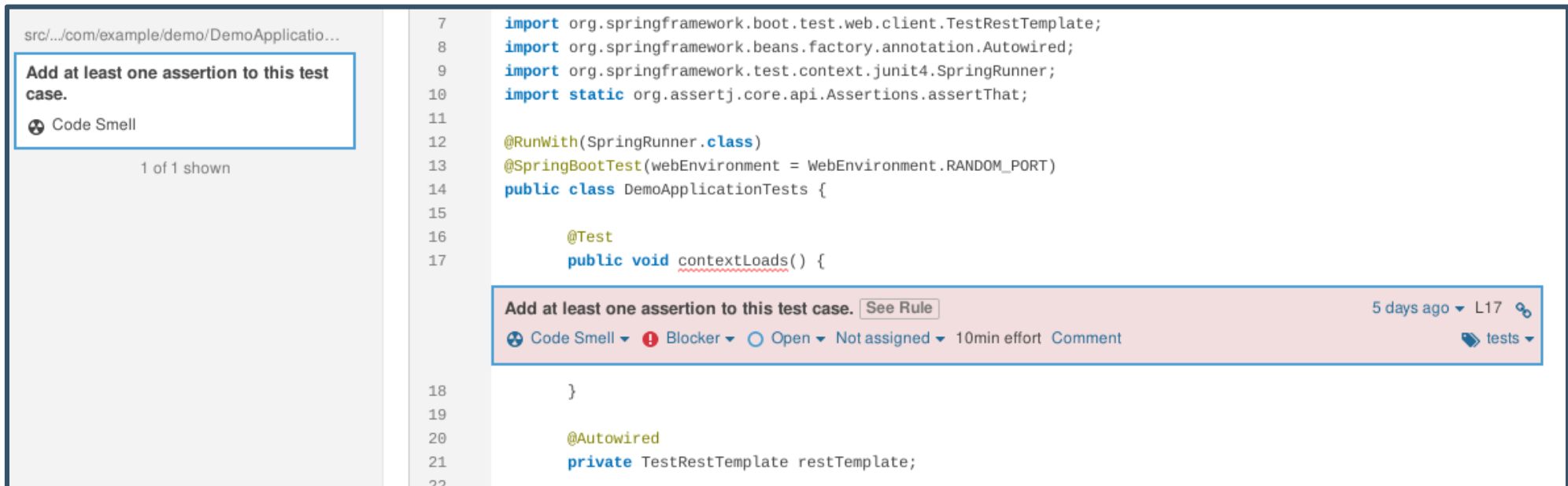


Detect Possible Code Issues sonarqube



Find bugs and performance issues that may not be easy to find

Detect Possible Code Issues sonarQube



The screenshot displays the SonarQube web interface for a Java code review. On the left, a sidebar shows a list of issues, with one 'Code Smell' issue highlighted. The main area shows the source code of a Java test class, `DemoApplicationTests`. A red box highlights a specific line of code, `contextLoads()`, which is marked as a 'Code Smell'. A tooltip message reads: 'Add at least one assertion to this test case.' Below the code, a detailed view of the issue is shown, including a 'See Rule' button, a list of filters (Code Smell, Blocker, Open, Not assigned, 10min effort), and a 'Comment' button. The issue is dated '5 days ago' and is located at 'L17'.

```
7 import org.springframework.boot.test.web.client.TestRestTemplate;
8 import org.springframework.beans.factory.annotation.Autowired;
9 import org.springframework.test.context.junit4.SpringRunner;
10 import static org.assertj.core.api.Assertions.assertThat;
11
12 @RunWith(SpringRunner.class)
13 @SpringBootTest(webEnvironment = WebEnvironment.RANDOM_PORT)
14 public class DemoApplicationTests {
15
16     @Test
17     public void contextLoads() {
18
19
20     @Autowired
21     private TestRestTemplate restTemplate;
22 }
```

Add at least one assertion to this test case.

Code Smell

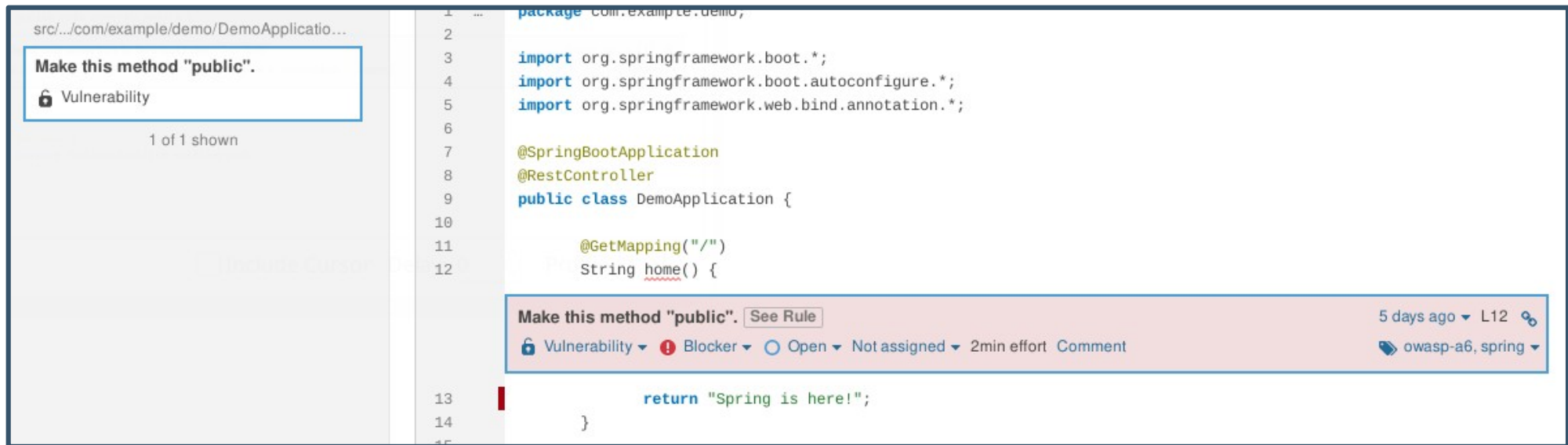
1 of 1 shown

Add at least one assertion to this test case. [See Rule](#) 5 days ago L17

Code Smell Blocker Open Not assigned 10min effort Comment tests

SonarQube detects possible Code Smells

Detect Possible Code Issues sonarQube



The screenshot displays the SonarQube web interface. On the left, a sidebar shows the project path `src/.../com/example/demo/DemoApplicatio...` and a list of issues. The first issue is highlighted with a blue box and contains the text "Make this method 'public'." and "Vulnerability". Below this, it says "1 of 1 shown".

The main area shows a code editor with the following Java code:

```
1 package com.example.demo;
2
3 import org.springframework.boot.*;
4 import org.springframework.boot.autoconfigure.*;
5 import org.springframework.web.bind.annotation.*;
6
7 @SpringBootApplication
8 @RestController
9 public class DemoApplication {
10
11     @GetMapping("/")
12     String home() {
13
14         return "Spring is here!";
15     }
16 }
```

At the bottom of the code editor, a red squiggly line under the `home()` method indicates an issue. A tooltip box is open over this line, showing the issue details:

- Issue title: "Make this method 'public'." with a "See Rule" link.
- Severity: "Vulnerability" (indicated by a lock icon).
- Other labels: "Blocker" (red exclamation mark), "Open" (blue circle), "Not assigned" (grey circle), "2min effort", and a "Comment" link.
- Metadata: "5 days ago", "L12", and a link icon.
- Rule ID: "owasp-a6, spring" with a dropdown arrow.

Don't worry about security invulnerability

Detect Possible Code Issues sonarqube

Filters Clear All Filters Bulk Change ↑ ↓ to select rules ← → to navigate 1 / 113 rules

Search for rules...

Language

▼ Type **BUG** Clear

- 🐛 Bug 113
- 🔒 Vulnerability 36
- 💡 Code Smell 212
- 🔥 Security Hotspot 30

> Tag

> Repository

> Default Severity

> Status

> Security Category

> Available Since

> Template

⬆️ ".equals()" should not be used to test the values of "Atomic" classes	Java	🐛 Bug	🔍 multi-threading	⌵	Deactivate
⬆️ "++" should not be used instead of "+="	Java	🐛 Bug		⌵	Deactivate
⚠️ "@Controller" classes that use "@SessionAttributes" must call "setComplete" on their "SessionStatus" objects	Java	🐛 Bug	🔍 spring	⌵	Deactivate
⬆️ "@NonNull" values should not be set to null	Java	🐛 Bug	🔍 cert, cwe	⌵	Deactivate
⚠️ "@SpringBootApplication" and "@ComponentScan" should not be used in the default package	Java	🐛 Bug	🔍 spring	⌵	Deactivate
⬆️ "BigDecimal(double)" should not be used	Java	🐛 Bug	🔍 cert	⌵	Deactivate
⬆️ "compareTo" results should not be checked for specific values	Java	🐛 Bug	🔍 unpredictable	⌵	Deactivate
⬆️ "compareTo" should not be overloaded	Java	🐛 Bug	🔍 pitfall	⌵	Deactivate
⬆️ "compareTo" should not return "Integer.MIN_VALUE"	Java	🐛 Bug		⌵	Deactivate
⬆️ "DefaultMessageListenerContainer" instances should not drop messages during restarts	Java	🐛 Bug	🔍 spring	⌵	Deactivate

Customizable rules allow you to fit guidelines for your projects

Getting Started

Using SonarCloud



Free for public projects

- **Unlimited** lines of code
- Project **visible** to everyone
- Access to **full feature set**
- **Restrict** access to work on your project

Paid for private projects

- Create **private projects**, priced per lines of code
- Restrict **complete** access to your project
- Pricing: 100k lines → 10€/month

- Login at <https://sonarcloud.io>
- You can either create an account or use single-sign on with GitHub, Bitbucket, Azure DevOps

Getting Started

Using SonarCloud



A screenshot of the SonarCloud website's landing page. The page has a white background with a large orange curved shape on the right side. In the top left corner is the "sonarcloud" logo with its icon. In the top right corner are links for "Pricing" and "Log in". The main heading is "Clean Code Rockstar Status" in orange and black. Below it is the text "Eliminate bugs and vulnerabilities. Champion quality code in your projects." followed by "Go ahead! Analyze your repo:". There are three buttons for "GitHub", "Bitbucket", and "Azure DevOps". Below these buttons is the text "Free for Open-Source Projects". On the right, there is a cartoon illustration of three people: a woman with blonde hair pointing up, a man with a beard and sunglasses wearing a red cap and an orange t-shirt that says "KEEP CALM AND WRITE CLEAN CODE", and a man with dark skin wearing a purple t-shirt that says "BUGS NO MORE".

<https://sonarcloud.io>

Prerequisites (Linux):

These additional configurations are needed due to the usage of an embedded ElasticSearch

```
$ sysctl -w vm.max_map_count=262144
$ sysctl -w fs.file-max=65536
$ ulimit -n 65536
$ ulimit -u 4096
```

Note: Properties set via `sysctl` are not persistent after restart of the host system.

Make sure to edit these values in `/etc/sysctl.d/...` as well.

Getting Started

SonarQube + PostgreSQL // Docker



docker-compose + bitnami image:

```
$ curl -LO https://raw.githubusercontent.com/bitnami/bitnami-docker-sonarqube/master/docker-compose.yml
$ docker-compose up
```

```
sonarqube_1 |
sonarqube_1 | Welcome to the Bitnami sonarqube container
sonarqube_1 | Subscribe to project updates by watching https://github.com/bitnami/bitnami-docker-sonarqube
sonarqube_1 | Submit issues and feature requests at https://github.com/bitnami/bitnami-docker-sonarqube/issues
sonarqube_1 |
postgresql_1 | postgresql 11:25:05.96
postgresql_1 | postgresql 11:25:05.98 Welcome to the Bitnami postgresql container
postgresql_1 | postgresql 11:25:05.99 Subscribe to project updates by watching https://github.com/bitnami/bitnami-d
postgresql_1 | postgresql 11:25:06.00 Submit issues and feature requests at https://github.com/bitnami/bitnami-d
postgresql_1 | postgresql 11:25:06.01 Send us your feedback at containers@bitnami.com
postgresql_1 | postgresql 11:25:06.02
postgresql_1 | postgresql 11:25:06.09 INFO ==> ** Starting PostgreSQL setup **
postgresql_1 | postgresql 11:25:06.37 INFO ==> Validating settings in POSTGRES*_* env vars..
postgresql_1 | postgresql 11:25:06.38 WARN ==> You set the environment variable ALLOW_EMPTY_PASSWORD=yes. For s
ag in a production environment.
postgresql_1 | postgresql 11:25:06.40 INFO ==> Loading custom pre-init scripts...
postgresql_1 | postgresql 11:25:06.43 INFO ==> Initializing PostgreSQL database...
postgresql_1 | postgresql 11:25:06.47 INFO ==> postgresql.conf file not detected. Generating it...
postgresql_1 | postgresql 11:25:06.53 INFO ==> pg_hba.conf file not detected. Generating it...
sonarqube_1 | nami INFO Initializing postgresql-client
```

Getting Started

Maven / Gradle goals



Maven (pom.xml)

```
<build>
  <plugins>
    <plugin>
      <groupId>org.sonarsource.scanner.maven</groupId>
      <artifactId>sonar-maven-plugin</artifactId>
      <version>3.6.0.1398</version>
    </plugin>
  </plugins>
</build>
```

Gradle (build.gradle)

```
plugins {
  id 'org.sonarqube' version '2.8'
}
```

Getting Started

Maven / Gradle goals



Maven

```
# Alternatively, write -D variables into your .m2/settings.xml
$ mvn sonar:sonar -Dsonar.host.url=<YOUR_URL> \
  -Dsonar.login=<YOUR_SONAR_TOKEN>
```

Gradle

```
# In ./gradle.properties
systemProp.sonar.host.url=<YOUR_URL>
systemProp.sonar.login=<YOUR_SONAR_TOKEN>

# In console (or IDE):
$ gradle sonarqube
```

Docker

Advantage of using Docker: Can be used in CI/CD

```
$ docker run -e SONAR_HOST_URL=<YOUR_SONAR_URL> \
-e SONAR_TOKEN=<YOUR_SONAR_TOKEN> \
--user="$(id -u):$(id -g)" \
-it \
-v "$PWD:/usr/src" \
sonarsource/sonar-scanner-cli
```

sonar-project.properties

```
sonar.projectName=<YOUR_PROJECT_NAME> # e. g. "Hello World"
sonar.projectKey=<YOUR_PROJECT_KEY> # e. g. com.example:hello-world
sonar.projectVersion=<YOUR_PROJECT_VERSION> # e. g. 1.0

sonar.sources=src
sonar.java.source=1.8
sonar.java.target=1.8
sonar.java.binaries=build/classes
```

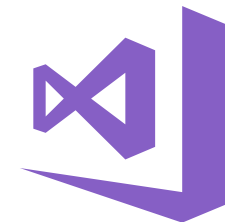
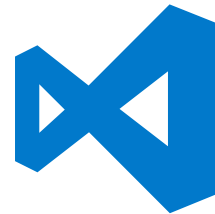
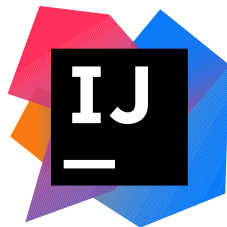
sonarlint



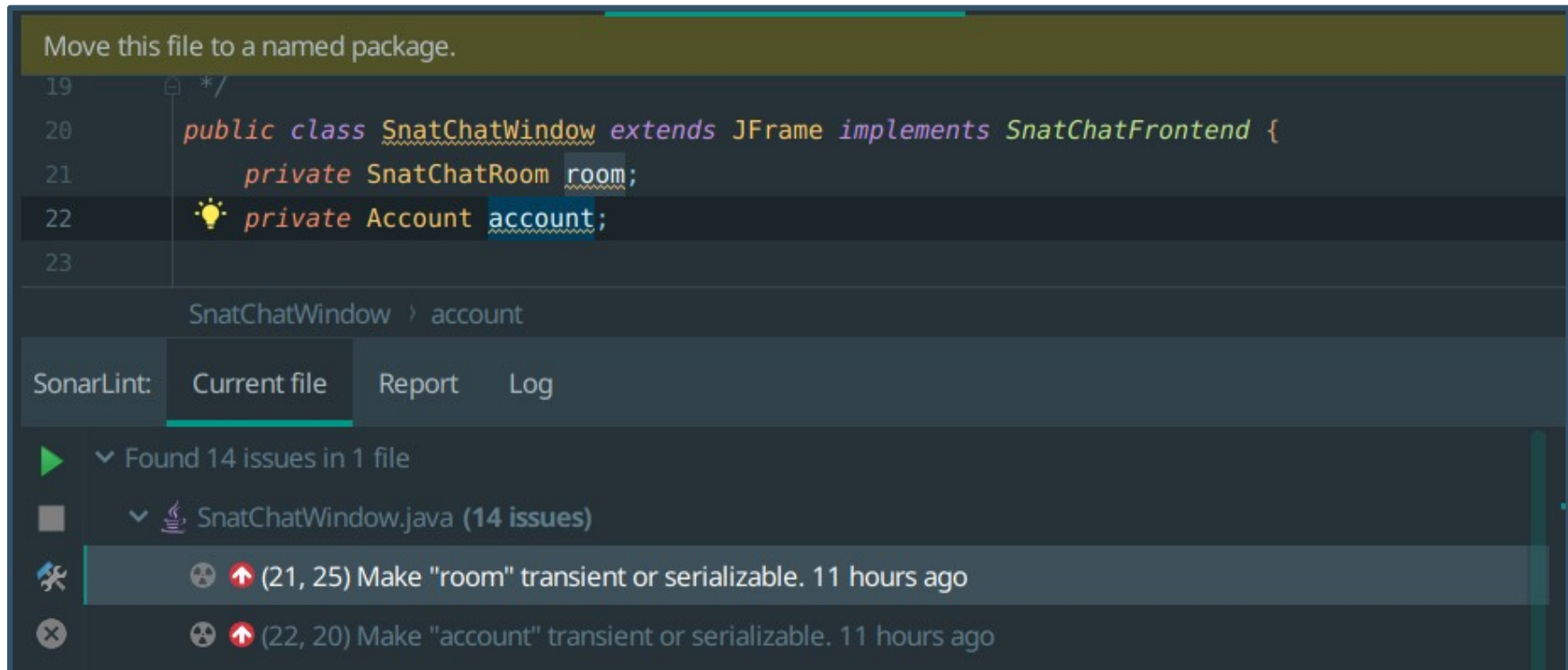
Fix issues before they exist

~ <https://www.sonarlint.org>

Available on:



IDE Integration



How it looks like in IntelliJ + Java

Custom Rules

Custom resources should be closed	Rule Template	Java	Bug	denial-of-service	▼
Track comments matching a regular expression	Rule Template	Java	Code Smell		▼
Track uses of disallowed classes	Rule Template	Java	Code Smell		▼
Track uses of disallowed constructors	Rule Template	Java	Code Smell		▼
Track uses of disallowed dependencies	Rule Template	Java	Code Smell	maven	▼
Track uses of disallowed methods	Rule Template	Java	Code Smell		▼

Predefined rule templates can be found in
Rules > *Filters* > *Templates* > *Show Templates Only*


Update Custom Rule

Name *

Key

squid:usage_system_exit_0

Severity **Status**

 Blocker Ready

Description *

Don't use it. Instead, I don't know. Make something up from your own.

[Markdown Help](#) : ***Bold*** ```Code``` * Bulleted point

MethodName

Name of the forbidden method

ClassName

Name of the class whose method is forbidden

ArgumentTypes

Save Cancel

Custom Rules

The screenshot displays the SonarQube interface with a code editor and a rule detail panel. The code editor shows a Java file `src/main/java/SnatChat.java` with lines 11, 12, and 13. Line 13 contains `System.exit(0);`, which is highlighted in yellow. A red vertical line indicates the current position in the code. The rule detail panel on the right shows the rule `squid:usage_system_exit_0` with the title `Usage of System.exit(0)`. The rule is categorized as `Vulnerability` and `Blocker`. It includes a `See Rule` button and a `Manual` button. The rule is marked as `Open` and `Not assigned`. The description of the rule states: `Don't use it. Instead, I don't know. Make something up from your own.`

src/main/java/SnatChat.java

Remove this forbidden call

Vulnerability

src/main/java/SnatChatRoom.java

Usage of System.exit(0)

Usage of System.exit(0)

Remove this forbidden call See Rule Manual 6 hours ago

Vulnerability Blocker Open Not assigned Comment

Usage of System.exit(0)

squid:usage_system_exit_0

Security Hotspot Blocker Main sources No tags Available Since Nov 13, 2019 SonarAnalyzer (Java) Custom Rule (Show Template)

Don't use it. Instead, I don't know. Make something up from your own.

Further advanced custom rule implementations are available. You can checkout <https://github.com/SonarSource/sonar-custom-rules-examples/tree/master/java-custom-rules> if you need an example

You can find an example project with Java and Gradle in my Git repository in folder "gradle-example"