



ABAP

Apex Apex

c c

C++

CloudFormation

COBOL COBOL

C# C#

S CSS

. .

X Flex

=GO

Go

∃ HTML

🆺 Java

JS JavaScript

Kotlin

Kubernetes

Objective C

PHP

PL/I

PL/SQL

Python s

RPG RPG

Ruby

Scala

Swift

Terraform

Text

TS TypeScript

VB VB.NET

VB6 VB6

XML XML



Go static code analysis

Unique rules to find Bugs, Security Hotspots, and Code Smells in your GO code

Hard-coded credentials are securitysensitive Security Hotspot **Cognitive Complexity of functions** should not be too high Code Smell String literals should not be duplicated Code Smell Functions should not be empty Code Smell All branches in a conditional structure should not have exactly the same implementation 📆 Bug "=+" should not be used instead of "+=" 📆 Bug Related "if/else if" statements should not have the same condition 📆 Bug Identical expressions should not be used on both sides of a binary operator 📆 Bug

All code should be reachable

Variables should not be self-assigned

Functions should not have identical

Two branches in a conditional structure should not have exactly the

"switch" statements should not have

same implementation

too many "case" clauses

📆 Bug

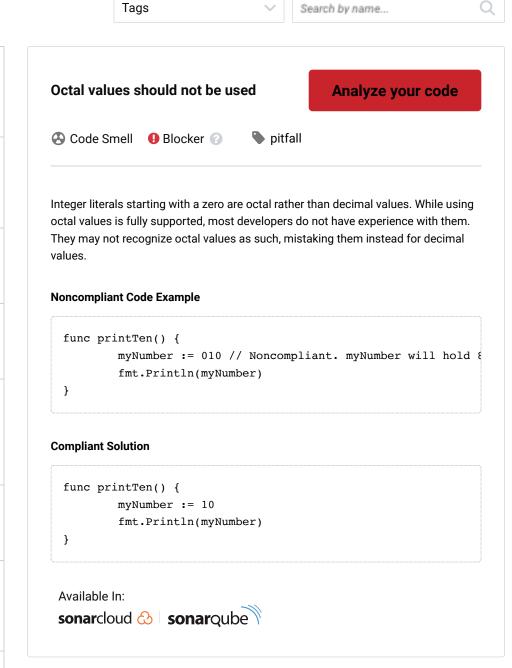
📆 Bug

implementations

Code Smell

Code Smell

Code Smell



© 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

Track uses of "FI	XME" tags
Code Smell	
Redundant pairs should be remov	
Code Smell	
Nested blocks of left empty	f code should not be
Code Smell	
Functions should parameters	d not have too many
☼ Code Smell	
Using hardcoded security-sensitiv	
Security Hots	spot
Multi-line commo	ents should not be
Code Smell	
Boolean checks inverted	should not be
Code Smell	
	nd function parameter omply with a naming
Code Smell	
Boolean literals s redundant	should not be
Code Smell	