

VB6

XML



Go static code analysis

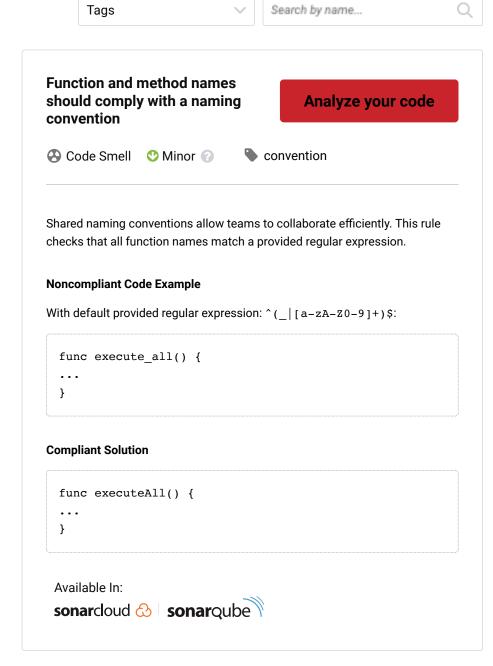
Unique rules to find Bugs, Security Hotspots, and Code Smells in your GO code

2

Hard-coded credentials are security- sensitive	Fun sho
Security Hotspot	con
Cognitive Complexity of functions should not be too high	< 0
Code Smell	Shar
String literals should not be duplicated	chec
Code Smell	None
Functions should not be empty	With
Code Smell	fu
All branches in a conditional structure should not have exactly the same implementation	Com
📆 Bug	fu
"=+" should not be used instead of "+="	
n Bug	
Related "if/else if" statements should not have the same condition	Ava SO
n Bug	
Identical expressions should not be used on both sides of a binary operator	© 20 S0 tra
📆 Bug	
All code should be reachable	
📆 Bug	
Variables should not be self-assigned	
Rug	
Functions should not have identical implementations	
Code Smell	

same implementation

Code Smell



Code Smell (29)

© 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

"switch" statements should not have too many "case" clauses
Track uses of "FIXME" tags
☼ Code Smell
Redundant pairs of parentheses should be removed
Nested blocks of code should not be left empty
Functions should not have too many parameters
Using hardcoded IP addresses is security-sensitive
Security Hotspot
Multi-line comments should not be empty
Boolean checks should not be inverted
◆ Code Smell
Local variable and function parameter names should comply with a naming convention
Boolean literals should not be redundant