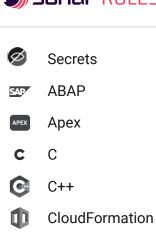
Code Smell 99





COBOL COBOL C#

∃ css

Flex Go

5 HTML

近 Java

Js JavaScript

Kotlin

Kubernetes

Objective C

PHP

PL/I PL/I

PL/SQL

Python

RPG RPG

Ruby

Scala

Swift

Terraform

Text

TypeScript

T-SQL

VB VB.NET

VB6 VB6

XML XML



Swift static code analysis

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

R Bug (14)

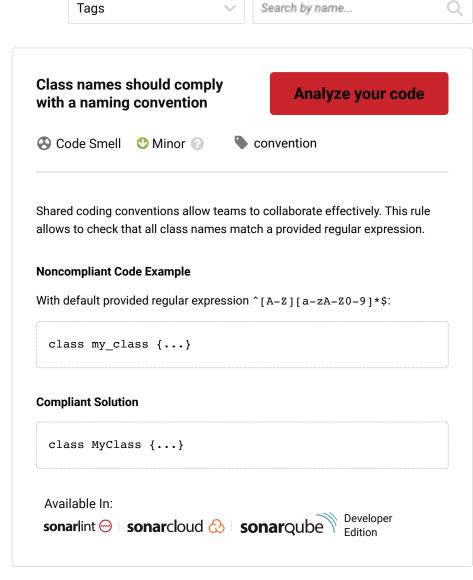
Hard-coded c	redentials	are security	/-
Security H	otspot		
Methods and be the same of capitalization	or differ on		ot
Code Sme	H		
Cipher algorit	hms shoul	d be robust	
⋒ Vulnerabil	ity		
Using weak h security-sens		orithms is	
Security H	otspot		
Cognitive Cor should not be		functions	
Code Sme	·II		
"try!" should r	ot be used	i	
Code Sme	<u> </u>		
String literals	should no	t be duplica	itec
Code Sme	411		
Functions and empty	d closures	should not	be
Code Sme	.		
Collection ele			
🛊 Bug			
Collection siz	es compai	risons shou	ld
🕦 Bug			
All branches i should not ha implementati	ve exactly		ure
🛊 Bug			

update their left operands

Precedence and associativity of standard operators should not be

Rug Bug

changed



Security Hotspot 3

© 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

∰ Bug
Return values from functions without side effects should not be ignored
∰ Bug
Related "if/else if" statements and "cases" in a "switch" should not have the same condition
Rug
Identical expressions should not be used on both sides of a binary operator
🖟 Bug
All code should be reachable
Rug
Loops with at most one iteration should be refactored
🖟 Bug
"IBInspectable" should be used correctly
Functions should not have identical implementations
Ternary operators should not be nested
Closure expressions should not be nested too deeply
Code Smell
Backticks should not be used around