



XML

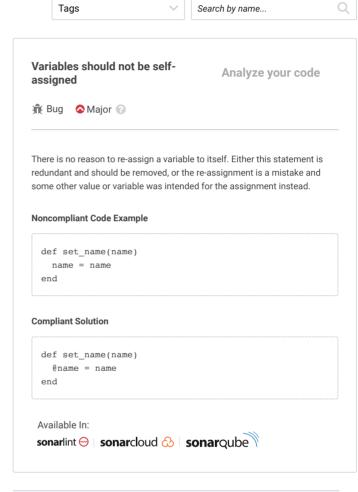


Ruby static code analysis

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

	Тас
☼ Code Smell	Veriable
Nested blocks of code should not be left empty	Variable assigne
Code Smell	∰ Bug
Functions should not have too many parameters	There is no redundant
Code Smell	some othe
Collapsible "if" statements should be merged	Noncomp
☼ Code Smell	def se
Using hardcoded IP addresses is security-sensitive	end
Security Hotspot	Compilant
Multi-line comments should not be empty	def se @nam end
Code Smell	
Boolean checks should not be inverted	Availabl sonar lir
☼ Code Smell	
Unused local variables should be removed	© 2008-20 SONAR, trademai
Code Smell	prope
function and block parameter names should comply with a naming convention	
Code Smell	
Class names should comply with a naming convention	
☼ Code Smell	
Method names should comply with a naming convention	
Code Smell Code Smell	

Track uses of "TODO" tags



♦ Code Smell ③3

© 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

Code Smell Track lack of copyright and license headers Code Smell Octal values should not be used Code Smell "case" statements should not be nested Code Smell Control flow statements "if", "for", "while", "until", "case" and "begin...rescue" should not be nested too deeply

← Code Smell