

- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- Code Smell
- C#
- CSS
- Flex
- Go
- HTML
- Java
- JavaScript
- Kotlin
- Objective C
- PHP
- PL/I
- PL/SQL
- Python
- RPG
- Ruby
- Scala
- Swift
- Terraform
- Text
- TypeScript
- T-SQL
- VB.NET
- VB6
- XML



PHP static code analysis

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

All rules 268 Vulnerability 40 Bug 51 Security Hotspot 33 Code Smell 144

Tags

Search by name...

Code Smell

A subclass should not be in the same "catch" clause as a parent class

Code Smell

Jump statements should not be redundant

Code Smell

"catch" clauses should do more than rethrow

Code Smell

"&&" and "||" should be used

Code Smell

Boolean checks should not be inverted

Code Smell

Local variables should not be declared and then immediately returned or thrown

Code Smell

Unused local variables should be removed

Code Smell

"switch" statements should have at least 3 "case" clauses

Code Smell

A "while" loop should be used instead of a "for" loop

Code Smell

Overriding methods should do more than simply call the same method in the super class

Code Smell

"empty()" should be used to test for emptiness

Variables should not be self-assigned

Analyze your code

Bug Major

There is no reason to re-assign a variable to itself. Either this statement is redundant and should be removed, or the re-assignment is a mistake and some other value or variable was intended for the assignment instead.

Noncompliant Code Example

```
public function setName($name) {
    $name = $name;
}
```

Compliant Solution

```
public function setName($name) {
    $this->name = $name;
}
```

Available In:

sonarlint sonarcloud sonarqube


© 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


Interface names should comply with a naming convention

 Code Smell

Return of boolean expressions should not be wrapped into an "if-then-else" statement

 Code Smell

Boolean literals should not be redundant

 Code Smell

Empty statements should be removed

 Code Smell