

- Secrets
- ABAP
- Apex
- C
- C++
- CloudFormation
- COBOL
- 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... 🔍

expressions should not contain only one character
Code Smell
Superfluous curly brace quantifiers should be avoided
Code Smell
Non-capturing groups without quantifier should not be used
Code Smell
WordPress option names should not be misspelled
Code Smell
WordPress options should not be defined at the end of "wp-config.php"
Code Smell
Constants should not be redefined
Code Smell
Regular expressions should not contain empty groups
Code Smell
Regular expressions should not contain multiple spaces
Code Smell
Single-character alternations in regular expressions should be replaced with character classes
Code Smell
Reluctant quantifiers in regular expressions should be followed by an expression that can't match the empty string
Code Smell
Character classes in regular expressions should not contain the same character twice
Code Smell

Methods should not be empty

Analyze your code

Code Smell

Critical ?

suspicious

- There are several reasons for a method not to have a method body:
- It is an unintentional omission and should be fixed to prevent unexpected behavior in production.
 - It is not yet, or never will be, supported. In this case an `UnsupportedOperationException` should be thrown.
 - The method is an intentionally-blank override. In this case a nested comment should explain the reason for the blank override.

Noncompliant Code Example

```
public function doSomething() {  
}  
  
public function doSomethingElse() {  
}
```

Compliant Solution

```
public function doSomething() {  
    // Do nothing because of X and Y.  
}  
  
public function doSomethingElse() {  
    throw new UnsupportedOperationException();  
}
```


Exceptions

Empty methods in abstract classes


```
abstract class Animal {  
    public function speak() {} // default implementation  
}
```

Available In:


sonarlint | sonarcloud | sonarqube

 Code Smell


Regular expressions should not be too complicated

 Code Smell

PHPUnit assertTrue/assertFalse should be simplified to the corresponding dedicated assertion

 Code Smell

Methods should not have identical implementations

 Code Smell

Functions should use "return" consistently

 Code Smell