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...

Two branches in a conditional structure should not have exactly the same implementation
Code Smell

Unused assignments should be removed
Code Smell

Method arguments with default values should be last
Code Smell

A reason should be provided when skipping a test
Code Smell

"__construct" functions should not make PHP 4-style calls to parent constructors
Code Smell

PHP 4 constructor declarations should not be used
Code Smell

Deprecated predefined variables should not be used
Code Smell

"switch" statements should not have too many "case" clauses
Code Smell

Classes should not have too many methods
Code Smell

Functions should not have too many lines of code
Code Smell

"for" loop stop conditions should be invariant

Assertions should not compare an object to itself

Analyze your code

Bug Major tests phpunit

Assertions comparing an object to itself are more likely to be bugs due to developer's carelessness.

This rule raises an issue when the actual expression matches the expected expression.

Noncompliant Code Example

```
assertEqual($a, $a); // Noncompliant
assertSame($a, $a); // Noncompliant
assertNotEqual($a, $a); // Noncompliant
assertNotSame($a, $a); // Noncompliant
```

Compliant Solution

```
assertEqual($expected, $a);
assertSame($expected, $a);
assertNotEqual($expected, $a);
assertNotSame($expected, $a);
```

Available In:
sonarlint | sonarcloud | sonarqube

Code Smell

Sections of code should not be commented out

Code Smell

Unused function parameters should be removed

Code Smell

Unused "private" methods should be removed

Code Smell

Functions should not contain too many return statements

Code Smell