

- 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...

be used to end lines	Code Smell
More than one property should not be declared per statement	Code Smell
The "var" keyword should not be used	Code Smell
"<?php" and "<?=" tags should be used	Code Smell
File names should comply with a naming convention	Code Smell
Comments should not be located at the end of lines of code	Code Smell
Local variable and function parameter names should comply with a naming convention	Code Smell
Field names should comply with a naming convention	Code Smell
Lines should not end with trailing whitespaces	Code Smell
Files should contain an empty newline at the end	Code Smell
Modifiers should be declared in the correct order	Code Smell
An open curly brace should be located at the beginning of a line	Code Smell

Assertions should not be made at the end of blocks expecting an exception

Analyze your code

Bug

Critical

tests pitfall phpunit

In PHPUnit, to test that an exception is thrown in a given piece of code, the `expectException()` methods or the `@expectedException` annotations can be used. For such a test to succeed, something in the test method has to throw an exception with the awaited properties. Having an assertion at the end of such a test method, means that, if the test succeeds, that assertion was never evaluated because an exception was thrown before getting to that point.

Noncompliant Code Example

```
public function testA() {
    $o = new MyClass();
    $this->expectException(\Exception::class);
    $o->doSomething();
    $this->assertTrue($o->hasProperty()); // Noncompliant -
}
```

Compliant Solution

```
public function testA() {
    $o = new MyClass();
    $this->expectException(\Exception::class);
    $o->doSomething();
}

public function testB() {
    $o = new MyClass();
    $this->assertTrue($o->hasProperty());
}
```

Available In:

sonarlint

sonarcloud

sonarqube


An open curly brace should be located at the end of a line

 Code Smell


Tabulation characters should not be used

 Code Smell

Method and function names should comply with a naming convention

 Code Smell

Creating cookies with broadly defined "domain" flags is security-sensitive

 Security Hotspot