
































-  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... 🔍

cause side-effects
 Bug
Increment (++) and decrement (--) operators should not be used in a method call or mixed with other operators in an expression
 Code Smell
Test methods should be discoverable
 Code Smell
Use of namespaces should be preferred to "include" or "require" functions
 Code Smell
Duplicate values should not be passed as arguments
 Code Smell
Configuration should not be changed dynamically
 Code Smell
Class constructors should not create other objects
 Code Smell
PHP parser failure
 Code Smell
The names of methods with boolean return values should start with "is" or "has"
 Code Smell
"php_sapi_name()" should not be used
 Code Smell
Classes should not have too many lines of code
 Code Smell

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

Analyze your code

 Code Smell

 Major ?

 tests phpunit

Testing equality or nullness with PHPUnit's assertTrue() or assertFalse() should be simplified to the corresponding dedicated assertion.

Noncompliant Code Example

```
assertTrue($a === $b);
assertTrue($a == $b);
assertTrue($a === null);
assertTrue($a !== null);
assertTrue($a !== $b);
assertTrue($a != $b);
assertFalse($a === $b);
assertFalse($a == $b);
```

Compliant Solution

```
assertEquals($a, $b);
assertSame($a, $b);
assertNull($a);
assertNotNull($a);
assertNotEquals($a, $b);
assertNotSame($a, $b);
assertNotEquals($a, $b);
assertNotSame($a, $b);
```

Available In:

sonarlint  | sonarcloud  | sonarqube 

Deprecated features should not be used

 Code Smell

Files should not contain inline HTML

 Code Smell

Files should contain only one top-level class or interface each

 Code Smell

Classes should not have too many fields

 Code Smell

Track uses of "NOSONAR" comments

 Code Smell