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...

Code Smell

Classes should not have too many lines of code

Code Smell

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

Statements should be on separate lines

Code Smell

Classes should not be coupled to too many other classes (Single Responsibility Principle)

Code Smell

"switch case" clauses should not have too many lines of code

Code Smell

Assignments should not be made from within sub-expressions

Code Smell

Files should not have too many lines of code

Code Smell

Classes named like "Exception" should extend "Exception" or a subclass

Analyze your code

Code Smell   Major   convention error-handling pitfall

Clear, communicative naming is important in code. It helps maintainers and API users understand the intentions for and uses of a unit of code. Using "exception" in the name of a class that does not extend `Exception` or one of its subclasses is a clear violation of the expectation that a class' name will indicate what it is and/or does.

Noncompliant Code Example

```
class FruitException {  // Noncompliant; this has nothi
  private $expected;
  private $unusualCharacteristics;
  private $appropriateForCommercialExploitation;
  // ...
}

class CarException {  // Noncompliant; the extends clau
  public function __construct(string message, Throwable
  // ...
```

Compliant Solution

```
class FruitSport {
  private $expected;
  private $unusualCharacteristics;
  private $appropriateForCommercialExploitation;
  // ...
}

class CarException extends Exception {
  public function __construct(string message, Throwable
  // ...
```

Available In:

sonarlint   sonarcloud   sonarqube

Lines should not be too long

Code Smell

HTTP response headers should not be vulnerable to injection attacks

Vulnerability

"sleep" should not be called

Vulnerability

Static members should be referenced with "static::"

Bug

"require_once" and "include_once" should be used instead of "require" and "include"