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

Regular expressions should be syntactically valid

Analyze your code

🐛 Bug   🔺 Critical ⑦   🏷 regex

Regular expressions have their own syntax that is understood by regular expression engines. Those engines will throw an exception at runtime if they are given a regular expression that does not conform to that syntax.

To avoid syntax errors, special characters should be escaped with backslashes when they are intended to be matched literally and references to capturing groups should use the correctly spelled name or number of the group.

To match a literal string, rather than a regular expression, either all special characters should be escaped or methods that don't use regular expressions should be used.

Noncompliant Code Example

```
preg_match("/([/", $input);
preg_replace("/[/", "{", $input);
preg_replace("/(\\w+-(\\d+)/", "1234", $input);
```

Compliant Solution

```
preg_match("/\\(\\[/", $input);
str_replace("([", "{", $input);
preg_replace("/(\\w+)-(\\d+)/", "1234", $input);
```

Available In:

sonarlint ⊙ | sonarcloud ⊛ | sonarqube ⌇

⊗ Code Smell

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