

PHP code coverage verifier

Tom Rochette <tom.rochette@coreteks.org>

December 21, 2025 — [77e1b28a](#)

1 Problem

I wrote some new code or edited code in my PHP application and I'd like to know if this change is covered by tests.

2 Solution

The approach I thought of to solve this problem was to write your change, run your tests (most likely using PHPUnit), then create a diff/patch of those changes and use this information to determine whether the lines that are part of the diff are covered. It should be relatively straightforward to automate the process of generating a patch from within a git repository as well as running PHPUnit within a project's directory (generating a `clover.xml` report). With this information in hand, it would be possible for a script to compute what is and isn't covered by tests in the changes that you are about to commit.

I wrote [PHP Code Coverage Verifier](#) which takes care of reading the PHPUnit output (a `clover.xml` file) and a generated patch file and produce a report of the lines that are and aren't covered.

An example of the script output is as follow:

```
php vendor/bin/php-code-coverage-verifier verify my-clover.xml my-diff.patch
Using clover-xml file: my-clover.xml
With diff file: my-diff.patch
```

Covered:

```
controller/admin/stocks.php line 15 - 21
controller/admin/stocks.php line 91 - 97
controller/search.php line 26 - 32
controller/search.php line 376 - 384
model/user.php line 34 - 41
model/user.php line 44 - 51
```

Not covered:

```
controller/account.php line 39 - 45
controller/admin/stocks.php line 27 - 33
controller/search.php line 36 - 42
controller/search.php line 187 - 193
model/user.php line 533 - 540
```

Ignored:

```
application/composer.json
```

Coverage: 40 covered (56.338%), 31 not covered (43.662%)

3 Reference

- <https://github.com/tomzx/php-code-coverage-verifier>