PHPUnit 12 für Contao/Symfony-Entwickler:innen

Sebastian Bergmann

Ich kann dir helfen

- PHPUnit einzuführen,
- deine Tests zu optimieren,
- deinen Entwicklungs-prozess zu verbessern und
- testbaren Code zu schreiben

thePHP.cc

Zuletzt haben wir uns vor 2 Jahren in Kiel gesehen.

Seitdem sind PHP 8.3 und PHP 8.4 erschienen.

Und PHPUnit 11 und PHPUnit 12.

Dann wollen wir doch mal schauen, was in den letzten beiden Jahren passiert ist!

PHP

- Asymmetric Visibility (PHP 8.4)
- Property Hooks (PHP 8.4)
- Lazy Objects (PHP 8.4)
- Pipe Operator (PHP 8.5)
- Typed Class Constants (PHP 8.3)
- `#[Deprecated]` Attribut (PHP 8.4)
- `#[NoDiscard]` Attribut (PHP 8.5)
- Verbesserungen in der DOM-Erweiterung (HTML 5 Support!) (PHP 8.4)
- `json_validate()` Funktion (PHP 8.3)
- RFC 3986 and WHATWG URL compliant API (PHP 8.5)
- Verbesserter JIT-Compiler (PHP 8.4)

Und sonst so?

- 10 Core Developer (vorher 6) von PHP Foundation finanziert (Januar 2024)
- Security Audit von Composer (Juni 2024)
- PHPStan 2.0 (November 2024)
- Security Audit von PHP (April 2025)
- The PHP Foundation unterstützt offiziell FrankenPHP (Mai 2025)
- 30 Jahre PHP! (Juni 2025)
- PIE 1.0 (Juni 2025)
- Infection 0.30 bindet (optional) PHPStan ein (Juli 2025)
- Packagist.org unterstützt Composer 1.x nicht mehr (September 2025)
- Offizielles PHP SDK für MCP (September 2025)

Major Version	PHP Compatibility	Release	End of Bugfix Support
PHPUnit 12	>= PHP 8.3	February 7, 2025	February 5, 2027
PHPUnit 11	>= PHP 8.2	February 2, 2024	February 6, 2026
PHPUnit 10	>= PHP 8.1	February 3, 2023	February 7, 2025
PHPUnit 9	>= PHP 7.3	February 7, 2020	February 2, 2024
PHPUnit 8	>= PHP 7.2	February 1, 2019	February 3, 2023

Bugfix Support	Life Support

Feature Sunsetting

- **Soft Deprecation**
  - `@deprecated` Annotation (falls möglich)
    - Vielleicht stattdessen in Zukunft `#[Deprecated]` Attribut?
  - Dokumentation

- **Hard Deprecation**
  - Deprecations werden angezeigt (falls möglich)

- **Entfernung**

- Beispiele
  - https://github.com/sebastianbergmann/phpunit/blob/10.5/DEPRECATIONS.md
  - https://github.com/sebastianbergmann/phpunit/blob/11.5/DEPRECATIONS.md

Beispiel: Test Doubles API

- `createTestProxy()`, `getMockForAbstractClass()`, `getMockFromWsdl()`, `getMockForTrait()`, `getObjectForTrait()`, `disableAutoload()`, `disableArgumentCloning()`, `allowMockingUnknownTypes()`, `addMethods()`, …
- Soft Deprecation in PHPUnit 10
- Hard Deprecation in PHPUnit 11
- Entfernt in PHPUnit 12

Beispiel: Test Doubles API

- Seit PHPUnit 12 klare Trennung zwischen Test Stubs und Mock Objects

- **Zur Erinnerung**
  - Services kapseln einen "computational process" und haben ein Interface
  - Wir nutzen eine echte Implementierung des Interfaces, wenn wir diese Implementierung testen wollen
  - Wir nutzen einen Test Stub des Interfaces, wenn wir eine Komponente testen wollen, die von der Schnittstelle abhängt
  - Wir nutzen ein Mock Objekt der Schnittstelle, wenn wir die Kommunikation zwischen zusammenarbeitenden Objekten testen wollen

Testing with(out) dependencies

- Effective use of test stubs and mock objects
- Writing robust and maintainable unit tests
- Avoiding common pitfalls

thePHP.cc

```
❯ ./tools/phpunit --filter TupleTest --no-progress --testdox
PHPUnit 12.3.14 by Sebastian Bergmann and contributors.

Runtime:       PHP 8.4.13
Configuration: /Users/sb/Work/OpenSource/raytracer/phpunit.xml

Time: 00:00.005, Memory: 27.77 MB

Tuple (SebastianBergmann\Raytracer\Tuple)
 ✔ A tuple with w=1.0 is a point
 ✔ A tuple with w=0.0 is a vector
 ✔ A tuple with w!=1.0 and w!=0.0 is neither a point nor a vector
 ✔ A vector can be added to a point
 ✔ A vector can be added to another vector
 .
 .
 ✔ The magnitude of vector (1.0, 0.0, 0.0) is calculated to be 1.0
 ✔ The magnitude of vector (0.0, 1.0, 0.0) is calculated to be 1.0
 ✔ The magnitude of vector (0.0, 0.0, 1.0) is calculated to be 1.0
 .
 .


OK (25 tests, 78 assertions)
```

TupleTest.php

```php
<?php declare(strict_types=1);
namespace SebastianBergmann\Raytracer;

// ...

#[CoversClass(Tuple::class)]
#[Small]
final class TupleTest extends TestCase
{
    public function test_a_vector_can_be_added_to_a_point(): void
    {
        // ...
    }

    // ...
}
```

```
 ✔ A vector can be added to a point
```

TupleTest.php

```php
<?php declare(strict_types=1);
namespace SebastianBergmann\Raytracer;

// ...

#[CoversClass(Tuple::class)]
#[Small]
final class TupleTest extends TestCase
{
    #[TestDox('A tuple with w=1.0 is a point')]
    public function test_a_tuple_with_w_equal_to_one_is_a_point(): void
    {
        // ...
    }

    // ...
}
```

```
 ✔ A tuple with w=1.0 is a point
```

TupleTest.php

```php
<?php declare(strict_types=1);
namespace SebastianBergmann\Raytracer;

// ...

#[CoversClass(Tuple::class)]
#[Small]
final class TupleTest extends TestCase
{
    #[DataProvider('magnitudeProvider')]
    #[TestDox(
      'The magnitude of vector ($x, $y, $z) is calculated to be $magnitude'
    )]
    public function test_the_magnitude_of_a_vector_can_be_calculated
    (float $magnitude, float $x, float $y, float $z): void
    {
        // ...
    }

    // ...
}
```

```
 ✔ The magnitude of vector (1.0, 0.0, 0.0) is calculated to be 1.0
```

ExampleTest.php

```php
// ...

public static function formatter
(int $minutes, DateTimeImmutable $a, DateTimeImmutable $b): string
{
    return sprintf(
        'Duration for %s - %s: %d minutes',
        $a->format('Y-m-d H:i'),
        $b->format('Y-m-d H:i'),
        $minutes,
    );
}

#[DataProvider('provider')]
#[TestDoxFormatter('formatter')]
public function testDurationCanBeCalculated
(int $minutes, DateTimeImmutable $a, DateTimeImmutable $b): void
{
    // ...
}

// ...
```

```
✔ Duration for 2025-08-28 19:10 - 2025-08-28 20:00: 50 minutes
```

Deprecations (and other issues)

src/FirstPartyClass.php

```php
<?php declare(strict_types=1);
namespace example;

use vendor\ThirdPartyClass;

final class FirstPartyClass
{
    public function method(): true
    {
        (new ThirdPartyClass)->method();

        @trigger_error(
            'deprecation in first-party code',
            E_USER_DEPRECATED
        );

        return true;
    }
}
```

vendor/ThirdPartyClass.php

```php
<?php declare(strict_types=1);
namespace vendor;

use example\FirstPartyClass;

final class ThirdPartyClass
{
    public function method(): void
    {
        @trigger_error(
            'deprecation in third-party code',
            E_USER_DEPRECATED
        );
    }

    public function anotherMethod(): true
    {
        return (new FirstPartyClass)->method();
    }
}
```

tests/FirstPartyClassTest.php

```php
<?php declare(strict_types=1);
namespace example;

use PHPUnit\Framework\TestCase;
use vendor\ThirdPartyClass;

final class FirstPartyClassTest extends TestCase
{
    public function testOne(): void
    {
        $this->assertTrue((new FirstPartyClass)->method());
    }

    public function testTwo(): void
    {
        $this->assertTrue((new ThirdPartyClass)->anotherMethod());
    }
}
```

phpunit.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<phpunit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation=
         "https://schema.phpunit.de/12.3/phpunit.xsd"
         bootstrap="vendor/autoload.php"
>
    <testsuites>
        <testsuite name="default">
            <directory>tests</directory>
        </testsuite>
    </testsuites>

    <source ignoreSuppressionOfDeprecations="true"
            ignoreDirectDeprecations="true"
            ignoreIndirectDeprecations="true">
        <include>
            <directory>src</directory>
        </include>
    </source>
</phpunit>
```

```
❯ ./tools/phpunit --display-deprecations
PHPUnit 12.3.14 by Sebastian Bergmann and contributors.

Runtime:       PHP 8.4.13
Configuration: /Users/sb/example/phpunit.xml

D.                                                                  2 / 2 (100%)

Time: 00:00.014, Memory: 12.00 MB

1 test triggered 1 deprecation:

1) /Users/sb/example/src/FirstPartyClass.php:12
deprecation in first-party code

Triggered by:

* example\FirstPartyClassTest::testOne
  /Users/sb/example/tests/FirstPartyClassTest.php:9

OK, but there were issues!
Tests: 2, Assertions: 2, Deprecations: 1.
```

```
❯ ./tools/phpunit --generate-baseline phpunit-baseline.xml
PHPUnit 12.3.14 by Sebastian Bergmann and contributors.

Runtime:       PHP 8.4.13
Configuration: /Users/sb/example/phpunit.xml

D.                                                                  2 / 2 (100%)

Time: 00:00.006, Memory: 8.00 MB

OK, but there were issues!
Tests: 2, Assertions: 2, Deprecations: 1.

Baseline written to /Users/sb/example/phpunit-baseline.xml.
```

```
❯ ./tools/phpunit --use-baseline phpunit-baseline.xml
PHPUnit 12.3.14 by Sebastian Bergmann and contributors.

Runtime:       PHP 8.4.13
Configuration: /Users/sb/example/phpunit.xml

..                                                                  2 / 2 (100%)

Time: 00:00.006, Memory: 8.00 MB

OK (2 tests, 2 assertions)

1 issue was ignored by baseline.
```

Diese Präsentation hat ein Zuhause im Web:

thePHP.cc

Ich möchte von euch lernen!

- 3. November 2025
- 16:00 Uhr (MEZ)
- 2 Stunden
- Kostenlos!

thePHP.cc

Danke!

https://phpunit.expert

sebastian@thephp.cc

@sebastian@phpc.social

Bildnachweise

- https://www.pexels.com/de-de/foto/dramatischer-vollmond-hinter-dem-kommunikationsturm-32690322
- https://2023.conference.contao.org/files/images/hero/hero-kiel.jpg
- https://www.pexels.com/de-de/foto/weisses-reihenboot-auf-gewasser-127160
- https://www.pexels.com/de-de/foto/roter-led-verkehrskegel-2743739