

Extending PHPUnit with its New Event System

Andreas Möller

**Andreas
Möller**

@localheinz

Agenda

1. Introduction
2. Extensibility
3. Test Listener
4. Hooks
5. Event System
6. Summary
7. Q & A

Extensibility

Test Listener

Test Listener

Available since

phpunit/phpunit:0.3

Test Listener
Provides a single
Interface

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

namespace PHPUnit\Framework;

use Throwable;

interface TestListener
{
    public function startTestSuite(TestSuite $suite): void;

    public function startTest(Test $test): void;

    public function addError(Test $test, Throwable $t, float $time): void;

    public function addWarning(Test $test, Warning $e, float $time): void;

    public function addFailure(Test $test, AssertionFailedError $e, float $time): void;

    public function addIncompleteTest(Test $test, Throwable $t, float $time): void;

    public function addRiskyTest(Test $test, Throwable $t, float $time): void;

    public function addSkippedTest(Test $test, Throwable $t, float $time): void;

    public function endTest(Test $test, float $time): void;

    public function endTestSuite(TestSuite $suite): void;
}
```


**Test Listener
Extension
Implementation**

```
in main
<?php

declare(strict_types=1);

namespace JohnKary\PHPUnit\Listener;

use PHPUnit\Framework\Test;
use PHPUnit\Framework\TestListener;
use PHPUnit\Framework\TestListenerDefaultImplementation;
use PHPUnit\Framework\TestSuite;

class SpeedTrapWithListener implements TestListener
{
    use TestListenerDefaultImplementation;

    public function endTest(Test $test, float $time): void
    {
    }

    public function startTestSuite(TestSuite $suite): void
    {
    }

    public function endTestSuite(TestSuite $suite): void
    {
    }
}

28:1 UTF-8 Php
```

Test Listener

Extension

Registration

```
<phpunit
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="https://schema.phpunit.de/8.5/phpunit.xsd"
  bootstrap="vendor/autoload.php"
>
  <listeners>
    <listener class="JohnKary\PHPUnit\Listener\SpeedTrapListener" />
  </listeners>

  <testsuites>
    <testsuite name="unit">
      <directory>test/Unit/</directory>
    </testsuite>
  </testsuites>
</phpunit>
```

```
<phpunit
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="https://schema.phpunit.de/8.5/phpunit.xsd"
  bootstrap="vendor/autoload.php"
>
<listeners>
  <listener class="JohnKary\PHPUnit\Listener\SpeedTrapListener">
    <arguments>
      <array>
        <element key="slowThreshold">
          <integer>500</integer>
        </element>
        <element key="reportLength">
          <integer>10</integer>
        </element>
        <element key="stopOnSlow">
          <boolean>>false</boolean>
        </element>
      </array>
    </arguments>
  </listener>
</listeners>

<testsuites>
  <testsuite name="unit">
    <directory>test/Unit</directory>
  </testsuite>
</testsuites>
</phpunit>
```

Test Listener

Deprecated since

phpunit/phpunit:8.0.0

Test Listener

Will be removed in

phpunit/phpunit:10.0.0

Hooks

Hooks

Available since

phpunit/phpunit:7.1.0

Hooks

Segregated

Interfaces

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

```
namespace PHPUnit\Runner;
```

```
interface Hook
```

```
{
```

```
}
```

```
<?php declare(strict_types=1);  
  
namespace PHPUnit\Runner;  
  
interface BeforeFirstTestHook extends Hook  
{  
    public function executeBeforeFirstTest(): void;  
}
```

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

```
namespace PHPUnit\Runner;
```

```
interface TestHook extends Hook
```

```
{
```

```
}
```

```
<?php declare(strict_types=1);  
  
namespace PHPUnit\Runner;  
  
interface AfterLastTestHook extends Hook  
{  
    public function executeAfterLastTest(): void;  
}
```

```
<?php declare(strict_types=1);  
  
namespace PHPUnit\Runner;  
  
interface BeforeTestHook extends TestHook  
{  
    public function executeBeforeTest(string $test): void;  
}
```

```
<?php declare(strict_types=1);  
  
namespace PHPUnit\Runner;  
  
interface AfterTestHook extends TestHook  
{  
    public function executeAfterTest(string $test, float $time): void;  
}
```



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

namespace PHPUnit\Runner;

interface AfterIncompleteTestHook extends TestHook
{
    public function executeAfterIncompleteTest(
        string $test,
        string $message,
        float $time
    ): void;
}
```

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

namespace PHPUnit\Runner;

interface AfterRiskyTestHook extends TestHook
{
    public function executeAfterRiskyTest(
        string $test,
        string $message,
        float $time
    ): void;
}
```

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

namespace PHPUnit\Runner;

interface AfterSkippedTestHook extends TestHook
{
    public function executeAfterIncompleteTest(
        string $test,
        string $message,
        float $time
    ): void;
}
```

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

namespace PHPUnit\Runner;

interface AfterSuccessfulTestHook extends TestHook
{
    public function executeAfterSuccessfulTest(string $test, float $time): void;
}
```

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

namespace PHPUnit\Runner;

interface AfterTestErrorHook extends TestHook
{
    public function executeAfterTestError(
        string $test,
        string $message,
        float $time
    ): void;
}
```

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

namespace PHPUnit\Runner;

interface AfterTestFailureHook extends TestHook
{
    public function executeAfterTestFailure(
        string $test,
        string $message,
        float $time
    ): void;
}
```

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

namespace PHPUnit\Runner;

interface AfterTestWarningHook extends TestHook
{
    public function executeAfterTestWarning(
        string $test,
        string $message,
        float $time
    ): void;
}
```

Hooks

Extension

Implementation


```
in main
<?php

declare(strict_types=1);

namespace JohnKary\PHPUnit\Extension;

use PHPUnit\Runner;

final class SpeedTrap implements
    Runner\AfterLastTestHook,
    Runner\AfterSuccessfulTestHook,
    Runner\BeforeFirstTestHook
{
    public function executeBeforeFirstTest(): void
    {
    }

    public function executeAfterLastTest(): void
    {
    }

    public function executeAfterSuccessfulTest(string $test, float $time): void
    {
    }
}
```

Hooks

Extension

Registration

<phpunit

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="https://schema.phpunit.de/8.5/phpunit.xsd"

bootstrap="vendor/autoload.php"

>

<extensions>

<extension class="JohnKary\PHPUnit\Extension\SpeedTrap" />

</extensions>

<testsuites>

<testsuite name="unit">

<directory>test/Unit/</directory>

</testsuite>

</testsuites>

</phpunit>

```
<phpunit
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="https://schema.phpunit.de/8.5/phpunit.xsd"
  bootstrap="vendor/autoload.php"
>
<extensions>
  <extension class="JohnKary\PHPUnit\Extension\SpeedTrap">
    <arguments>
      <array>
        <element key="slowThreshold">
          <integer>500</integer>
        </element>
        <element key="reportLength">
          <integer>10</integer>
        </element>
      </array>
    </arguments>
  </extension>
</extensions>

<testsuites>
  <testsuite name="unit">
    <directory>test/Unit</directory>
  </testsuite>
</testsuites>
</phpunit>
```

Hooks

Deprecated since

phpunit/phpunit:8.5.23

Hooks

Will be removed in

phpunit/phpunit:10.0.0

Event System

Event System

History



Event System

Events

Event System

Event Interface

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

namespace PHPUnit\Event;

interface Event
{
    public function telemetryInfo(): Telemetry\Info;

    public function asString(): string;
}
```

Event System

Event

Implementation

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

namespace PHPUnit\Event\Test;

use PHPUnit\Event\Event;
use PHPUnit\Event\Telemetry;

final class AssertionFailed implements Event
{
    public function telemetryInfo(): Telemetry\Info
    {
    }

    public function value(): mixed
    {
    }

    public function count(): int
    {
    }

    public function message(): string
    {
    }

    public function asString(): string
    {
    }
}
```

Event System

Event Tracer

```
<?php declare(strict_types=1);  
  
namespace PHPUnit\Event\Tracer;  
  
use PHPUnit\Event\Event;  
  
interface Tracer  
{  
    public function trace(Event $event): void;  
}
```


Event System

Event Subscriber

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

```
namespace PHPUnit\Event;
```

```
interface Subscriber
```

```
{
```

```
}
```

```
<?php declare(strict_types=1);  
  
namespace PHPUnit\Event\Test;  
  
use PHPUnit\Event\Subscriber;  
  
interface AssertionFailedSubscriber extends Subscriber  
{  
    public function notify(AssertionFailed $event): void;  
}
```

Event System

Event Facade

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

```
namespace PHPUnit\Event;
```

```
/**
```

```
 * @internal
```

```
 */
```

```
final class Facade
```

```
{
```

```
    public static function registerSubscribers(Subscriber ...$subscribers): void
```

```
    {
```

```
    }
```

```
    public static function registerSubscriber(Subscriber $subscriber): void
```

```
    {
```

```
    }
```

```
    public static function registerTracer(Tracer\Tracer $tracer): void
```

```
    {
```

```
    }
```

```
    public static function emitter(): Emitter
```

```
    {
```

```
    }
```

```
}
```

Event System

Event Emitter

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

```
namespace PHPUnit\Event;
```

```
/**
```

```
 * @internal
```

```
 */
```

```
interface Emitter
```

```
{
```

```
    // ...
```

```
}
```

Event System

Extension

Implementation


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

namespace PHPUnit\Runner\Extension;

use PHPUnit\TextUI\Configuration\Configuration;

interface Extension
{
    public function bootstrap(
        Configuration $configuration,
        Facade $facade,
        ParameterCollection $parameters
    ): void;
}
```

Event System
Extension Facade

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

namespace PHPUnit\Runner\Extension;

use PHPUnit\Event\Subscriber;
use PHPUnit\Event\Tracer\Tracer;

final class Facade
{
    public function registerSubscribers(Subscriber ...$subscribers): void
    {
    }

    public function registerSubscriber(Subscriber $subscriber): void
    {
    }

    public function registerTracer(Tracer $tracer): void
    {
    }
}
```

Event System

Extension

Registration

<phpunit

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="https://schema.phpunit.de/10.0/phpunit.xsd"

bootstrap="vendor/autoload.php"

>

<extensions>

<bootstrap class="JohnKary\PHPUnit\Extension\SpeedTrapExtension" />

</extensions>

<testsuites>

<testsuite name="unit">

<directory>test/Unit/</directory>

</testsuite>

</testsuites>

</phpunit>

<phpunit

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="https://schema.phpunit.de/10.0/phpunit.xsd"

bootstrap="vendor/autoload.php"

>

<extensions>

<bootstrap class="JohnKary\PHPUnit\Extension\SpeedTrapExtension">

<parameter name="slowThreshold" value="500"/>

<parameter name="reportLength" value="10"/>

</bootstrap>

</extensions>

<testsuites>

<testsuite name="unit">

<directory>test/Unit/</directory>

</testsuite>

</testsuites>

</phpunit>

Event System

Will be released with

phpunit/phpunit:10.0.0

Event System Migration

github.com/johnkary/phpunit-speedtrap/pull/98

The screenshot shows a GitHub pull request page for the repository 'johnkary/phpunit-speedtrap'. The pull request title is 'Enhancement: Update extension to use event system of phpunit/phpunit:10.0.0 #98'. It is a draft pull request created by 'localheinz' to merge 1 commit into 'johnkary:master' from 'localheinz:feature/phpunit-10'. The page includes navigation tabs for Code, Issues (15), Pull requests (8), Discussions, Actions, Projects, Wiki, Security, and Insights. The pull request details show 0 conversations, 1 commit, 1 check, and 10 files changed. A net change of +217 lines and -128 lines is indicated. A comment from 'localheinz' is visible at the bottom, with a 'Contributor' role and 'No reviews' status.

johnkary / **phpunit-speedtrap** Public

Unwatch 13 Fork 58 Starred 747

Code Issues 15 **Pull requests 8** Discussions Actions Projects Wiki Security Insights

Enhancement: Update extension to use event system of phpunit/phpunit:10.0.0 #98

Edit Code

Draft localheinz wants to merge 1 commit into `johnkary:master` from `localheinz:feature/phpunit-10`

Conversation 0 Commits 1 Checks 1 Files changed 10 +217 -128

localheinz commented 9 hours ago Contributor

Reviewers
No reviews

Event System

Details

github.com/sebastianbergmann/phpunit/issues/4676

The screenshot shows a web browser window displaying a GitHub issue page. The address bar shows the URL `github.com/sebastianbergmann/phpunit/issues/4676`. The GitHub navigation bar includes a search box, 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. The repository name is 'sebastianbergmann / phpunit' (Public). Action buttons include 'Sponsor', 'Unwatch 331', 'Fork 2.1k', and 'Starred 18.7k'. A secondary navigation bar shows 'Code', 'Issues 103', 'Pull requests 30', 'Actions', 'Security', and 'Insights'. The issue title is 'Event System for extending PHPUnit #4676', with 'Edit' and 'New issue' buttons. A status bar indicates the issue is 'Closed' and was opened by 'sebastianbergmann' on May 20, 2021, with 8 comments. The main content area shows a comment from 'sebastianbergm...' with the text 'History and Background' and 'The TestListener Interface'. The right sidebar shows 'Assignees' (No one—assign yourself) and 'Labels' (feature/events, type/enhancement).

github.com/sebastianbergmann/phpunit/issues/4676

Search or jump to... Pull requests Issues Marketplace Explore

sebastianbergmann / phpunit Public

Sponsor Unwatch 331 Fork 2.1k Starred 18.7k

<> Code Issues 103 Pull requests 30 Actions Security Insights

Event System for extending PHPUnit #4676

Edit New issue

Closed sebastianbergmann opened this issue on May 20, 2021 · 8 comments

sebastianbergm... commented on May 20, 2021 · edited

Owner

History and Background

The TestListener Interface

Assignees
No one—assign yourself

Labels
feature/events type/enhancement

Summary

1. Introduction
2. Extensibility
3. Test Listener
4. Hooks
5. Event System
6. Summary
7. Q & A

Q & A

Thank you!

am@localheinz.com

twitter.com/@localheinz

github.com/localheinz

localheinz.com