

Содержание

- Новости
- Идея сайта
- ▼ Статьи
 - ▼ Операционные системы
 - Linux
 - QNX
 - ▼ Программирование
 - ▼ Java
 - Примеры
 - **PHP**
 - Web
 - Zend Framework
 - Тестирование
 - ▼ СУБД
 - MySQL
 - Oracle
 - Наши работы
 - Ссылки

Метки

.Net CSS Drupal Eclipse Fedora
HTTP Java JavaDB Linux
MySQL Oracle Oracle Linux
PHP PHPUnit PL/SQL QNX
Smarty SWT Twitter UAC UTF8
Web Windows7 XML xUnit
Zend Framework Безопасность
Книги ОСи разное Регулярные
выражения **СУБД**
Тестирование

PHPUnit. Часть 05 Организация тестов

Статьи -> Программирование -> PHP

PHPUnit. Часть 05 Организация тестов

v:1.0 30.03.2010

Перевод статьи [Chapter 7. Organizing Tests.](#)

Автор: Sebastian Bergmann

Перевод: Петрелевич Сергей

Предисловие переводчика

Эта статья продолжает серию переводов официальной документации по PHPUnit на русский язык.

Часть 1, Часть 2, Часть 3, Часть 4,

Комбинируемость - это одна из целей PHPUnit (см. [Цели PHPUnit](#)).

Мы хотим иметь возможность запускать несколько тестов отдельно или все вместе, например, все тесты для всего проекта целиком или все тесты для всех классов только одного компонента, который является частью проекта или тесты только для одного класса.

PHPUnit поддерживает различные способы организации тестов и объединения их в тестовый набор. В этой статье рассматривается наиболее распространенный подход.

Составление набора тестов при помощи файловой системы

Пожалуй самый простой способ составить набор тестов заключается в хранении файлов всех исходных текстов в специальном каталоге для тестов. PHPUnit может автоматически найти и запустить тесты, рекурсивно обойдя такой каталог.

Давайте посмотрим набор тестов для библиотеки [Object_Freezer](#). Посмотрев на структуру каталогов, мы заметим, что структура тестовых классов - System Under Test (SUT), к каталоге `Tests` полностью повторяет структуру классов в каталоге `Object`.

```
Object Tests |-- Freezer |-- Freezer | |--
HashGenerator | |-- HashGenerator | | `--
NonRecursiveSHA1.php | | `--
NonRecursiveSHA1Test.php | |-- HashGenerator.php |
| | |-- IdGenerator | |-- IdGenerator | | `--
UUID.php | | `-- UUIDTest.php | |-- IdGenerator.php
| | | |-- LazyProxy.php | | | |-- Storage | |--
Storage | | `-- CouchDB.php | | `-- CouchDB | | | |
|-- WithLazyLoadTest.php | | | | `--
WithoutLazyLoadTest.php | |-- Storage.php | |--
StorageTest.php | `-- Util.php | `-- UtilTest.php
`-- Freezer.php `-- FreezerTest.php
```

Чтобы запустить все тесты библиотеки, нам всего лишь надо перейти в каталог с тестами PHPUnit и выполнить команду запуска:

```
phpunit Tests PHPUnit 3.4.2 by Sebastian Bergmann.
.....
..... 60 / 75 ..... Time: 0 seconds
OK (75 tests, 164 assertions)
```

Чтобы выполнить все тесты, которые декларированы в классе `Object_FreezerTest (Tests/FreezerTest.php)`, надо выполнить следующую команду:

Мой блог в Живом Журнале
Мой блог на Хабрахаб

Рассылки Subscribe.Ru
Килограмм килобайтов
ваш e-mail <input type="text"/> <input type="button" value="OK"/>
Подписаться письмом



```
phpunit Tests/FreezerTest PHPUnit 3.4.2 by
Sebastian Bergmann. ....
Time: 0 seconds OK (28 tests, 60 assertions)
```

Для более точного управления тестами можно использовать переключатель `--filter`:

```
phpunit --filter testFreezingAnObjectWorks Tests
PHPUnit 3.4.2 by Sebastian Bergmann. . Time: 0
seconds OK (1 test, 2 assertions)
```

Примечание

Недостаток этого подхода заключается в том, что мы не можем управлять последовательностью выполнения тестов. Это может привести к проблемам зависимости тестов, см. [раздел "Зависимости тестов"](#).

Составление набора тестов при помощи XML-конфигурации

Конфигурационный XML-файл PHPUnit ([Приложение C](#)) тоже может быть применен для составления тестового набора. [Пример 7.1](#) показывает как добавить все классы `*Test`, которые будут найдены в файлах `*Test.php` во время рекурсивного обхода каталога `Tests`.

Пример 7.1: Составление тестового набора при помощи XML-конфигурации

```
<phpunit <testsuites> <testsuite
name="Object_Freezer"> <directory>Tests</directory>
</testsuite> </testsuites> </phpunit>
```

Примечание

У этого подхода тот же недостаток, что и в предыдущем способе - мы не можем управлять последовательностью выполнения тестов. Это может привести к проблемам зависимости тестов, см. [раздел "Зависимости тестов"](#).

В качестве альтернативы, мы можем четко прописать порядок запуска тестов:

Пример 7.2: Составление тестового набора при помощи XML-конфигурации

```
<phpunit <testsuites> <testsuite
name="Object_Freezer">
<file>Tests/Freezer/HashGenerator/NonRecursiveSHA1Test.php</file>
<file>Tests/Freezer/IdGenerator/UUIDTest.php</file>
<file>Tests/Freezer/UtilTest.php</file>
<file>Tests/FreezerTest.php</file>
<file>Tests/Freezer/StorageTest.php</file>
<file>Tests/Freezer/Storage/CouchDB/WithLazyLoadTest.php</file>
<file>Tests/Freezer/Storage/CouchDB/WithoutLazyLoadTest.php</file>
</testsuite> </testsuites>
</phpunit>
```

Использование класса TestSuite

В состав PHPUnit входит класс `PHPUnit_Framework_TestSuite`. Этот класс позволяет организовать набор тестов в иерархическом порядке.

[Пример 7.3](#) демонстрирует класс `AllTests`, расположенный на вершине иерархии проекта, в состав которого входит пакет `Package`.

Пример 7.3: Родительский класс иерархии - AllTests

```
<?php
require_once 'PHPUnit/Framework.php';

require_once 'Package/AllTests.php';
// ...
```

```

class AllTests
{
    public static function suite()
    {
        $suite = new
PHPUnit_Framework_TestSuite('Project');

        $suite-
>addTest(Package_AllTests::suite());
        // ...

        return $suite;
    }
}
?>
* This source code was highlighted with Source Code Highlighter.

```

Класс AllTests включает классы пакета Package_AllTests, которые в свою очередь включают тестовые классы.

Пример 7.4: Класс Package_AllTests

```

<?php
require_once 'PHPUnit/Framework.php';

require_once 'Framework/ClassTest.php';
// ...

class Package_AllTests
{
    public static function suite()
    {
        $suite = new
PHPUnit_Framework_TestSuite('Package');

        $suite-
>addTestSuite('Package_ClassTest');
        // ...

        return $suite;
    }
}
?>
* This source code was highlighted with Source Code Highlighter.

```

Package_ClassTest - это типовой класс тестов, который наследует класс PHPUnit_Framework_TestCase.

- Выполнение phpunit AllTests в каталоге Tests приведет к запуску всех тестов.
- Выполнение phpunit AllTests в каталоге Tests/Package приведет к запуску тестов классов Package_*.
- Выполнение phpunit ClassTest в каталоге Tests/Package приведет к запуску тестов класса Package_Class (которые объявлены в классе Package_ClassTest).
- Выполнение phpunit --filter testSomething ClassTest в каталоге Tests/Package приведет к запуску теста testSomething класса Package_ClassTest.

Классы PHPUnit_Framework_TestSuite предлагают два шаблонных метода setUp() и tearDown(), которые вызываются до первого и после последнего теста тестового набора.

Пример 7.5: Класс MySuite

```

<?php
require_once 'MyTest.php';

class MySuite extends
PHPUnit_Framework_TestSuite

```

```

{
    public static function suite()
    {
        return new MySuite('MyTest');
    }

    protected function setUp()
    {
        print __METHOD__ . "\n";
    }

    protected function tearDown()
    {
        print __METHOD__ . "\n";
    }
}
?>

```

* This source code was highlighted with Source Code Highlighter.

Тестовый класс `MyTest`, добавленный в набор тестовых методов `MySuite` в [Примере 7.5](#) содержит два тестовых метода `testOne()` и `testTwo()`. В результатах работы теста видно в каком порядке вызываются функции `testOne()` и `testTwo()`, `setUp()` и `tearDown()`.

```

MySuite::setUp() MyTest::setUp() MyTest::testOne()
MyTest::tearDown() MyTest::setUp()
MyTest::testTwo() MyTest::tearDown()
MySuite::tearDown()

```

Метод `setUp()` класса `PHPUnit_Framework_TestSuite` сохраняет переменные в `$this->sharedFixture`. Эти данные доступны во всех тестах этого набора как `$this->sharedFixture` (см. [раздел "Совместное использование тестового окружения"](#)).

Примечание

Методы класса `TestSuite` `setUp()` и `tearDown()` будут вызваны даже если набор тестов не содержит тесты.

Метки: [PHP](#) [Web](#) [PHPUnit](#) [Тестирование](#)

Комментарии.

Внимание.

Комментировать могут только зарегистрированные пользователи.
Возможно использование следующих HTML тегов: `<a>`, ``, `<i>`, `
`.

Аноним 01.06.2010 11:17:56

Решилась заказать шкафы купе гардеробные расскажите что то,такой ассортимент стал такой огромный запуталась абсолютно

Пожалуйста [авторизуйтесь](#) или [зарегистрируйтесь](#).

