

Test your code like a pro PHPUnit in practice

Sebastian Marek, Software Architect



PHP North West Conference
06/10/12 : Manchester, UK

Me

- a Pole living in Sheffield
- over 12 years in development
- big fan of process automation
- TDD and CI
- occasionally contributes to open source software



You?

- ⦿ Who are you?
- ⦿ What's your typical environment you work in?
- ⦿ What's your experience in unit testing?
- ⦿ What are your expectations?

Agenda

- ⌚ Preparations / Setup
- ⌚ Simple test case
- ⌚ Fixing first failure
- ⌚ PHPUnit CLI options
- ⌚ Assertions
- ⌚ phpunit.xml file
- ⌚ Asserting exceptions
- ⌚ Asserting PHP errors
- ⌚ Asserting output
- ⌚ Using data providers
- ⌚ Generating code coverage report

Resources

- ⌚ WIFI: phpnw12-phpunit
- ⌚ hosts file: 192.168.254.55 phpnw12-tutorial
- ⌚ <http://phpnw12-tutorial>

Rescue plan

- ④ \$ git clone http://phpnw12-tutorial/phpnw12-tutorial.git
- ④ \$ cd phpnw12-tutorial
- ④ \$ git reset --hard
- ④ \$ git pull origin exercise-<number>

Exercise 1

Simple test case

- ⦿ repository structure
 - ⦿ src & tests
- ⦿ naming conventions
 - ⦿ Tutorial.php and TutorialTest.php
- ⦿ extends \PHPUnit_Framework_TestCase

Exercise 1 - Simple test case

Test first!

- namespace `PhpNw12\Tests\Workshop`
- class `TutorialTest`
- method `testGreetingsReturnWelcomeMessage()`
- `$ phpunit tests/Workshop/TutorialTest.php`

Exercise 1 - Simple test case

Code later!

- namespace **PhpNw12\Workshop**
- class **Tutorial**
- method **greetings()** - empty for now
- \$ phpunit tests/Workshop/TutorialTest.php

Exercise 1 - Simple test case

Fix the code!

- method **greetings()** - implement the functionality
- \$ phpunit tests/Workshop/TutorialTest.php

Exercise 1 - Simple test case

Another test

- `@test` annotation
- method `GetAttendeesReturnsListOfNames()`
- `assertInternalType()`
- `$ phpunit tests/Workshop/TutorialTest.php`

Exercise 1 - Simple test case

Implementation

- method **getAttendees()** - return empty array
- \$ phpunit tests/Workshop/TutorialTest.php

Exercise 2 - Fixing the failure

Refactoring

- add `$_attendees` property
- method `getAttendees()` return `$_attendees`
- \$ `phpunit tests/Workshop/TutorialTest.php`

Exercise 2 - Fixing the failure

Refactoring continues

- ➊ initialize `$_attendees` in the constructor
- ➋ `$ phpunit tests/Workshop/TutorialTest.php`

Exercise 2 - Fixing the failure

Fix broken code

- assign default value in the constructor
- \$ phpunit tests/Workshop/TutorialTest.php

Exercise 3

PHPUnit CLI options

- --colors
- --testdox
- --debug
- --filter

Exercise 4 - More assertions

Testing booleans - part 1

- new test
`testTutorialHasNoPlacesLeftWhenMoreThen3Attendees()`
- `assertFalse()`
- new method `arePlacesLeft()`
- \$ phpunit tests/Workshop/TutorialTest.php

Exercise 4 - More assertions

Testing booleans - part 2

- new test

`testTutorialHasPlacesLeftWhenLessThan3Attendees()`

- `assertTrue()`

- `$ phpunit tests/Workshop/TutorialTest.php`

Exercise 4 - More assertions

Testing numbers

- new test
`testTutorialIsNotFullWhenItNotExceedsMaximumCapacity()`
- `assertGreaterThanOrEqual()`, `assertLessThanOrEqual()`, `assertNotNull()`
- \$ phpunit tests/Workshop/TutorialTest.php

Exercise 4 - More assertions

Refactor to add functionality

- new const **MAX_CAPACITY**
- new method **getNumberOfAttendees()**
- refactor **arePlacesLeft()** to use the above
- \$ phpunit tests/Workshop/TutorialTest.php

Exercise 4 - More assertions

Testing variable types

- new test `testRoomIsAvailable()`
- `assertInstanceOf()`
- new class `Room`
- new property `$_room` initialized in the constructor
- \$ phpunit tests/Workshop/TutorialTest.php

Exercise 4 - More assertions

Testing arrays

- new test `testAttendeeGotAddedToList()`
- `assertContains()`
- new test
`testGetAttendeesReturnCorrectNumberOfAttendees()`
- `assertCount()`
- `$ phpunit tests/Workshop/TutorialTest.php`

Exercise 5 - phpunit.xml

- phpunit.xml.dist vs phpunit.xml

Exercise 6

Asserting exceptions

- new test

`testAddAttendeeThrowsExceptionWhenAddingNewPersonToFullTutorial()`

- `setExpectedException()`

- new method `addAttendee()`

- \$ phpunit

Exercise 6

Asserting exceptions

Refactor to add functionality

- ⌚ refactor `arePlacesLeft()`
- ⌚ refactor method `addAttendee()`
- ⌚ \$ phpunit

Exercise 6

Asserting exceptions

Using annotations

- ⌚ **@expectedException**
- ⌚ **@expectedExceptionMessage**
- ⌚ **@expectedExceptionCode**
- ⌚ **\$ phpunit**

Exercise 6

Asserting exceptions

Using try...catch

- catching exceptions using conventional methods
- **fail()**
- \$ phpunit

Exercise 7

Asserting errors

Testing PHP errors

- new test
 - `testInitiatingTutorialWithWrongParamThrowsError()`
- PHP errors converted to `PHPUnit_Framework_Error`
- \$ phpunit

Exercise 7

Asserting errors

Testing PHP warnings/notices

- new method `Room::includeDependencies()`
- new test
`testIncludeDependenciesThrowsWarningForMissingFiles()`
- PHP warnings converted to
`PHPUnit_Framework_Error_Warning`
- PHP notices converted to `PHPUnit_Framework_Error_Note`
- \$ phpunit

Exercise 8

Asserting output

- new test `testDisplayShowsGreetingsMessage()`
- new method `displaySummary()`
- `expectOutputString()`
- `$ phpunit`

Exercise 9

Data providers

- new provider

`provideListOfAttendeesNotExceedingMaximumTutorialCapacity()`

- modify test

`testTutorialIsNotFullWhenItExceedsMaximumCapacity()`

- @dataProvider

- \$ phpunit

Exercise 10

Code coverage

- `--coverage-html`
- `forceCoversAnnotation`
- `@covers`
- `$ phpunit --coverage-html=/tmp/phpnw12-tutorial`

Q & A

<https://joind.in/talk/view/7069>