

Codesception - тестируем по-человечески

Михаил Боднарчук



Обо мне



1. Веб разработчик из Киева
2. Пишу на PHP, Ruby, CoffeeScript
3. Разрабатываю open-source продукты:
Codeception, Robo, AspectMock
4. ~~По выходным летаю на пепелаце~~
(сломался)



О тестировании

1. Стоит ли вообще тестировать?
2. Что это за страшные слова: TDD, BDD, Test Doubles, Story BDD, Spec
BDD, Stubs, Mocks, Spies, Fakes, Fixtures?
3. С чего начать тестирование?

Начинают писать тесты

1. Когда после деплоя ничего не работает
2. Когда после мерджа всё поломалось
3. Когда надоело проверять API вручную
4. Когда постоянно приходится всё проверять вручную

Подходы к тестированию

1. **Прагматический подход** - Test After
2. **TDD** - Test First (Is TDD Dead?)
3. **BDD** - улучшенный TDD
 - Спец BDD - внутренние спецификации
 - Story BDD - требования заказчика



Best Practices

1. Тест состоит из **условия, действия, проверки**
2. Отделить дополнительный код от теста
3. Отделить конфигурацию от теста
4. Делать тесты читабельными (и компактными)

С чего начать тестирование?

- Установить PHPUnit. Найти самый простой изолированный и редко используемый класс. Написать на него тест.
- Установить Codeception. Написать тест проверяющий работу приложения в целом

0 Codeception

1. Современный фреймворк для тестирования в PHP
2. Прост в установке, продвигает best practices тестирования
3. Поддерживает приемочные, функциональные и юнит тесты
4. Более 20 модулей, включая **Selenium, Symfony, Yii, Laravel...**
5. Основан на **PHPUnit, Symfony Components, Guzzle**
6. <http://codeception.com>

Немножко истории...

- Проект назывался TestGuy
- За основу был взят PHPUnit 3.5
- Плагин для symfony 1.4
- Первый релиз в январе 2012
- ...всё это время я почти ничего не знал про тестирование

Пирамида тестирования



✓ Приемочные тесты

Функциональные тесты

Юнит тесты

Простой тест

```
01. $I = new AcceptanceTester($scenario);  
02. $I->amOnPage('/login');  
03. $I->fillField('email', 'myemail@test.com');  
04. $I->fillField('password', 'mysecurepassword');  
05. $I->click('#loginForm input[type=submit]');  
06. $I->seeCurrentUrlIs('/dashboard');  
07. $I->see('Welcome back', '.notice');
```

Читабельные тесты

мысленно уберем все спецсимволы...

01. I = new AcceptanceTester
02. I amOnPage '/login'
03. I fillField 'email', 'myemail@test.com'
04. I fillField 'password', 'mysecurepassword'
05. I click '#loginForm input[type=submit]'
06. I seeCurrentUrlIs '/dashboard'
07. I see 'Welcome back', '.notice'

Автодополнение

```
$I->click('Add new blog');  
$I->see('Add new blog');  
$I->
```

m amOnPage (\$page)	TestGuy
m attachFile (\$field, \$filename)	TestGuy
m checkOption (\$option)	TestGuy
m click (\$link)	TestGuy
m dontSee (\$text)	TestGuy
m dontSeeCheckboxIsChecked (\$checkbox)	TestGuy
m dontSeeInField (\$field, \$value)	TestGuy
m dontSeeLink (\$text)	TestGuy
m fillField (\$field, \$value)	TestGuy
m see (\$text)	TestGuy
m seeCheckboxIsChecked (\$checkbox)	TestGuy

Press again to see more variants. >>

Умная работа с локеаторами



01. `$I->click('Login');`

02. `$I->click('#loginForm input[type=submit]');`

03. `$I->click('.*[@id='loginForm']/input[@type='submit']');`

04. `$I->click(['name' => 'submit']);`

PageObjects

```
01. $I->click(LoginPage::$submitButton);  
02. LoginPage::of($I)  
03.     ->login('myemail@test.com', 'mysecurepassword');
```


StepObjects

```
01. $I = new AcceptanceTester\UserSteps($scenario);  
02. $I->login('myemail', 'mysecurepassword');  
03. $I->see('Welcome back', '.notice');
```

- Selenium WebDriver
- PhpBrowser
- Функциональные тесты PHP фреймворков

Selenium WebDriver

- Современный клиент WebDriver для PHP
- Разрабатывается в Facebook
- Максимально близко к официальным биндингам Java
- github.com/facebook/php-webdriver

Расширяем возможности



```
01. function clickGreenButton() {  
02.     $this->webDriver->findElement(  
03.         WebDriverBy::cssSelector('#green button')  
04.     )->click();;  
05. }  
06. // в тесте:  
07. $I->clickGreenButton();
```

Environments

- Последовательный запуск тестов с различной конфигурацией
- Например, тестирование в различных браузерах
- Например, тестирование на разных СУБД

Приемочные тесты

✓ Функциональные тесты

Юнит тесты

Простой тест

```
01. $I = new FunctionalTester($scenario);  
02. $I->amOnPage('/login');  
03. $I->fillField('email', 'myemail@test.com');  
04. $I->fillField('password', 'mysecurepassword');  
05. $I->click('#loginForm input[type=submit]');  
06. $I->seeCurrentUrlIs('/dashboard');  
07. $I->see('Welcome back', '.notice');  
08. $I->seeInSession('user_name', 'me');
```

Фреймворки

- Symfony 2 + Doctrine 2
- Yii, Yii2
- Laravel
- Zend Framework 1,2
- Phalcon


```
01. $I = new ApiTester($scenario);  
02. $I->wantTo('get pending articles via REST API');  
03. $I->amHttpAuthenticated('service_user', '123456');  
04. $I->sendGET('articles', ['type' => 'pending']);  
05. $I->seeResponseCodeIs(200);  
06. $I->seeResponseIsJson();  
07. $I->seeResponseContains('article: { title: "Article 1"}');
```

Приемочные тесты

Функциональные тесты

✓ Юнит тесты

Юнит тесты

- Нативная поддержка тестов PHPUnit
- Дополнительные хэлперы для работы с БД
- Запуск вместе с функциональными и приемочными

```
01. function testSavingUser() {  
02.     $user = new User();  
03.     $user->setName('Miles');  
04.     $user->save();  
05.     $this->assertEquals('Miles', $user->getFullName());  
06.     $this->tester->seeInDatabase('users', ['name' => 'Miles']);  
07. }
```

Codeception\AspectMock



```
01. $this->assertEquals('users', User::tableName());  
02. test::double('User', ['tableName' => 'fake_users']);  
03. $this->assertEquals('fake_users', User::tableName());
```

github.com/Codeception/AspectMock

Specify + Verify

```
01. $this->specify("returns full name", function() {  
02.     $this->user->firstName = 'Michael';  
03.     $this->user->lastName = 'Bodnarchuk';  
04.     verify($this->user->getFullName())  
05.         ->equals('Michael Bodnarchuk');  
06. });
```







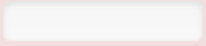
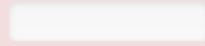

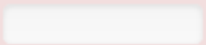
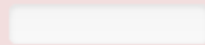
github.com/Codeception/Specify + github.com/Codeception/Verify

Общие возможности

Code Coverage



[/var/www/stage.test/src](#) / [Codeception](#) / [BackofficeBundle \(Dashboard\)](#)

	Code Coverage					
	Lines			Functions and Methods		
Total		68.21%	103 / 151		45.00%	9 / 20
 Controller		76.87%	103 / 134		52.94%	9 / 17
 DependencyInjection		0.00%	0 / 13		0.00%	0 / 2
 BackofficeBundle.php		0.00%	0 / 4		0.00%	0 / 1

Legend

Low: 0% to 35% **Medium:** 35% to 70% **High:** 70% to 100%

Generated by [PHP_CodeCoverage 1.2.6](#) using [PHP 5.4.9](#) at Fri Jan 4 18:35:21 EET 2013.

Отладочная информация

```
Scenario:
* I am going to submit a post as a regular user
* I am on page "/"
* I click "Submit a Post"
=> [GET] http://127.0.0.1:5000/?page_id=5 []
=> [Cookies] []
=> [Status] 200
=> [Headers] {"host":["127.0.0.1:5000"],"connection":["close"],"x-powered-by":["PHP/5.4.9-
text/html; charset=UTF-8"]}
* I fill field "Your Name", "Michael"
FAILED
```

- логирование
- скриншот страницы перед фейлом

Continuous Integration

- JUnit (XML) репорт
 - HTML Report
-

- Jenkins
- Bamboo
- Travis CI
- ...

Документация

- Обучающие материалы на официальном сайте
- Полная информация по API модулей
- [И даже видеокурс](#)

Codeception 2.0

Принципы

- Работает? - не трожь
- Сложно? - упрости
- Не нужно? - убери

Нововведения

- PHP 5.4, PHPUnit 4.x, Guzzle 4.x
- Guy => Tester
- Расширение для перезапуска упавших тестов
- Friends для мультимессионного тестирования
- Паралельный запуск тестов

Паралельный запуск тестов

- Разбиваем тесты на группы
- Выполняем тесты в X паралельных процессах
- Объединяем результаты

Параллельный запуск тестов



- Используем [Robo](#) task runner
- Подключаем задачи из `codeception/robo-paracept`
- Реализуем тестирование под свой проект

Codeception это...

- мультiformатный фреймворк для тестирования
- использует возможности PHPUnit
- **набор модулей** для тестирования веб приложений
- open-source проект с лицензией MIT
- спонсируется компанией 2Amigos

Спасибо за внимание

1. Михаил Боднарчук, автор
2. Мой скайп: davert.ua
3. Сайт проекта: codeception.com
4. И его twitter: @codeception
5. © <?= date('Y') ?>