

Mink, PHPUnit und PageObject

PHP Unconference

Benjamin Eberlei
Qafoo GmbH
21. September 2013



Helping people to create high quality web applications.

<http://qafoo.com>

@qafoo

- ▶ Trainings
- ▶ Consulting
- ▶ Tools

Outline

PHPUnit and Mink

- ▶ Web acceptance test framework
- ▶ Abstracts browser emulations / controllers
 - ▶ Goutte
 - ▶ Sahi
 - ▶ Selenium / Selenium 2
 - ▶ Zombie.js
- ▶ <http://mink.behat.org>

Why not Behat?

- ▶ Gherkin sentences are yet another abstraction
- ▶ If you don't have PM that use it, why not stick with one tool?
- ▶ Mink sentences are not readable by PMs

PHPUnit TestCase for Mink

```
1 <?php
2
3 abstract class ShopwareMinkTestCase extends PHPUnit_Framework_TestCase
4 {
5     protected $driver;
6     protected $session;
7
8     public function setUp()
9     {
10         parent::setUp();
11
12         $this->driver = new \Behat\Mink\Driver\GoutteDriver();
13         $this->session = new \Behat\Mink\Session($this->driver);
14     }
15
16     protected function visit($relativePath)
17     {
18         if (!isset($_SERVER['SHOPWARE_BASE'])) {
19             $_SERVER['SHOPWARE_BASE'] = 'http://localhost:8000';
20         }
21         $this->session->visit($_SERVER['SHOPWARE_BASE'] . $relativePath);
22
23         return $this->session->getPage();
24     }
25 }
```

Our first Mink Test

```
1 <?php
2
3 class ProductSearchTest extends ShopwareMinkTestCase
4 {
5     public function testSearchStrandtuch ()
6     {
7         $page = $this->visit ( '/search?sSearch=Strandtuch ' );
8
9         $this->assertContains (
10             'Zu "Strandtuch" wurden 5 Artikel gefunden ',
11             $page->getContent ()
12         );
13     }
14 }
```

Symfony CSS Selector Component

- ▶ Convert CSS Selector to XPath Expressions
- ▶ Use XPath with DOMDocument

```
1 <?php
2
3 namespace Symfony\Component\CssSelector\CssSelector;
4
5 $xpathId = CssSelector::toXPath('#content');
6 $xpathTitle = CssSelector::toXPath('#content_h1');
7 $priceCells = CssSelector::toXPath('.block-prices_tbody_td');
```


Using CSS Selectors

```
1 <?php
2
3 class ProductSearchTest extends ShopwareMinkTestCase
4 {
5     public function testSearchStrandtuch()
6     {
7         $page = $this->visit('/search?sSearch=Strandtuch');
8
9         $this->assertEquals(
10             'Zu "Strandtuch" wurden 5 Artikel gefunden!',
11             $page->find('css', '.result_box')->getText()
12         );
13     }
14 }
```

Click Links and Submit Forms

- ▶ Use the Session like a real browser
- ▶ Click on links that are present on current page
- ▶ Submit forms that are present on current page
- ▶ Benefits
 - ▶ Tests in terms of user-interaction
 - ▶ No URL hardcoding in the tests

Submit Forms

```
1 <?php
2
3 class ProductSearchTest extends ShopwareMinkTestCase
4 {
5     public function testSearchStrandtuch()
6     {
7         $page = $this->visit('/');
8
9         $page->fillField('submit_search_btn', 'Strandtuch');
10        $page->pressButton('submit_search_btn');
11
12        $this->assertEquals(
13            'Zu "Strandtuch" wurden 5 Artikel gefunden',
14            $page->find('css', '.result_box')->getText()
15        );
16    }
17 }
```

Click Links

```
1 <?php
2
3 class ProductCategoryTest extends ShopwareMinkTestCase
4 {
5     public function testGenussweltenKategorie ()
6     {
7         $page = $this->visit ( '/' );
8
9         $page->clickLink ( 'Genusswelten' );
10
11        $this->assertContains (
12            'Special_Finish_Lagerkorn_X.O. ',
13            $page->getContent ()
14        );
15    }
16 }
```

Execute Javascript

Invoke javascript in Browser from Mink

```
1 $session->evaluateScript('alert("Hello World!");');
```

Page Objects

Within your web app's UI there are areas that your tests interact with. A Page Object simply models these as objects within the test code.

- ▶ Keep browser tests maintainable
- ▶ Solves Problems:
 - ▶ UI, Workflows change all the time
 - ▶ Tests break often



THANK YOU

Rent a quality expert
qafoo.com