Magento®
eCommerce Platform for Growth

Unit Tests für Magento

Eine praktische Einführung

Fabian Schmengler

8. Magento-Stammtisch Aachen, 21.6.2012

Vorstellung

- **Fabian Schmengler**
- Webentwicklung seit 2005
- SGH IT seit 2008
  - Gesellschafter, Prokurist, Entwickler
- Magento-Modulentwicklung seit 2011
- Kontakt
  - fschmengler@sgh-it.eu

Unit Tests

- Testen einer Programm-Einheit (Klasse)
- Vergleich von Ausgabe mit **erwarteter** Ausgabe
- **Automatisierbar**
- **Isoliert**
  - Simulation anderer Klassen („Mock Object“)
  - Umgebung bei jedem Durchlauf gleich („Fixture“)
    - z.B. Test-Datenbank

PHPUnit

- De-facto Standard
- xUnit Framework-Familie (JUnit, CppUnit etc.)
- Einfache Installation über PEAR Installer

  → http://www.phpunit.de/
- IDE Integration (Test per Mausklick)
  - Eclipse PTI, Zend Studio, Netbeans, PhpStorm ...

Begriffe

- **Fixture:** Zustand der Umgebung, Kontext
- **Test Case:** eine Test-Einheit (Klasse)
  - Kann mehrere Tests (Methoden) enthalten
- **Test Suite:** Zusammenfassung von Test Cases
- **Assertion:** Vergleich mit erwartetem Wert
- **Mock:** Vom Test kontrolliertes Dummy-Objekt

Beispiel: Test Case

```php
<?php
class HelloWorldTest extends PHPUnit_Framework_TestCase
{
    private $helloWorldApp;

    public static function setUpBeforeClass()
    {
    }
    public static function tearDownAfterClass()
    {
    }
    protected function setUp()
    {
        $this->helloWorldApp = new HelloWorld();
    }
    protected function tearDown()
    {
    }
    public function testHelloWithLanguage()
    {
        $out = $this->helloWorldApp->hello('de');
        $this->assertEquals('Hallo Welt', $out, 'hello() should speak german.');
    }
}
```

Fixture

Test

erwartet Wert Nachricht

EcomDev PHPUnit

http://www.ecomdev.org/

- Extension: Magento PHPUnit Testing
- Magento-spezifisches Fixture-Management
  - Mage_App
  - „saubere“ Testdatenbank
  - Test-Daten für Entities in YAML-Dateien je Test
- Mock Objects
  - Rückgabe von Mocks in Mage::getModel() etc.
- Gute Dokumentation: http://bit.ly/mR6uKc

EcomDev PHPUnit

http://www.ecomdev.org/

- Basis-Test Case für
  - **Models, Helpers, Blocks**
- Spezielle Test Cases für
  - **Controllers:** Testen von Funktion und Layout
  - **Config:** Testen der config.xml (!)
- Separation von Tests und Test-Daten
  - Datenbank-Zustand (fixtures/*.yaml)
  - Eingabedaten (dataprovider/*.yaml)
  - Erwartete Daten (expectations/*.yaml)

Model Test mit Testdaten

- Beispiel: ID-Kodierung für Short URLs
  - Test wird mit jedem Parametersatz aus dem Data Provider aufgerufen

```
-
  - 1
-
  - 10
-
  - 63
-
  - 64
-
  - 65
-
  - 100
-
  - 1000
-
  - 10000
-
  - 100000
-
  - 821054
-
  - 900021
```

```
/**
 * @link h[illegible]
 * @link
 *   http[illegible]
 *   [illegible] 23-1, [illegible]
 * @test
 * @loadExpectations
 * @dataProvider dataProvider
 */
public function testEncode($id)
{
    $this->assertEquals(
        $this->expected("id-$id")->getEncoded(),
        $this->_model->encode($id)
    );
}
```

```
id-1:
  encoded: B
id-10:
  encoded: K
id-63:
  encoded: _
id-64:
  encoded: BA
id-65:
  encoded: BB
id-100:
  encoded: Bk
id-1000:
  encoded: Po
id-10000:
  encoded: CcQ
id-100000:
  encoded: Yag
id-821054:
  encoded: DIc-
id-900021:
  encoded: Dbu1
```

DB-Fixtures

- Werden automatisch in Test-DB übernommen
  - parent::setUp() nicht vergessen!
- Fixture-Datei *innerhalb von Test Case* wiederverwendbar
- Typen von Einträgen:
  - Website, Group, Store (scope)
  - Entities (eav)
  - Flat Tables (tables)
  - Konfiguration (config)
- Unvollständige Daten erlaubt

```yaml
eav:
  catalog_product:
    -
      entity_id: 1
      stock:
        qty: 100
        is_in_stock: 1
      website_ids:
        - aubu_website
      category_ids:
        - 2 #default
      type_id: bundle
      has_options: 1
      required_options: 1
      sku: test
      name: Test
      status: 1 # enabled
      price_view: 1
      visibility: 4 # catalog & search
    -
      entity_id: 2
```

Config Test

```
public function testObserver()
{
    $this->assertEventObserverDefined(
        'frontend', 'core_block_abstract_to_html_after',
        SGH_ShortUrls_Model_Observer_Block::MODEL,
        'onCoreBlockAbstractToHtmlAfter', 'shorturls', 'Observer'
    );
}
```

Observer Assertion

```
public function testRouterConfig()
{
    $this->assertConfigNodeHasChildren(
        'default/web/routers', 'router config exists'
    );
    $this->assertConfigNodeHasChild(
        'default/web/routers', 'shorturls_product',
        'shorturls_product router config'
    );
    $this->assertConfigNodeValue(
        'default/web/routers/shorturls_product/area', 'frontend'
        'shorturls_product router area: fr
    );
    $this->assertConfigNodeValue(
        'default/web/routers/shorturls_pro
        SGH_ShortUrls_Controller_Router_Sh
        'shorturls_product router router c
    );
}
```

Testen konkreter Knoten in config.xml

Weitere fertige Assertions für:

- Class Aliase
- Modulkonfiguration
- Einbindung von Layout-Datei

Controller Test

- Dispatch-Methoden
- Zahlreiche Assertions für
  - Routing
  - Layout
  - Response Headers
  - Response Body
- Nützlich für Integrationstests
- Kein vollständiger Ersatz für funktionale Tests!

```
$this->dispatch('customer/account/login');
$this->assertRequestRoute('customer/account/login');
$this->assertLayoutHandleLoaded('customer_account_login');
$this->assertLayoutHandleNotLoaded('cms_index_index');
$this->assertResponseBodyContains('login or create');
```

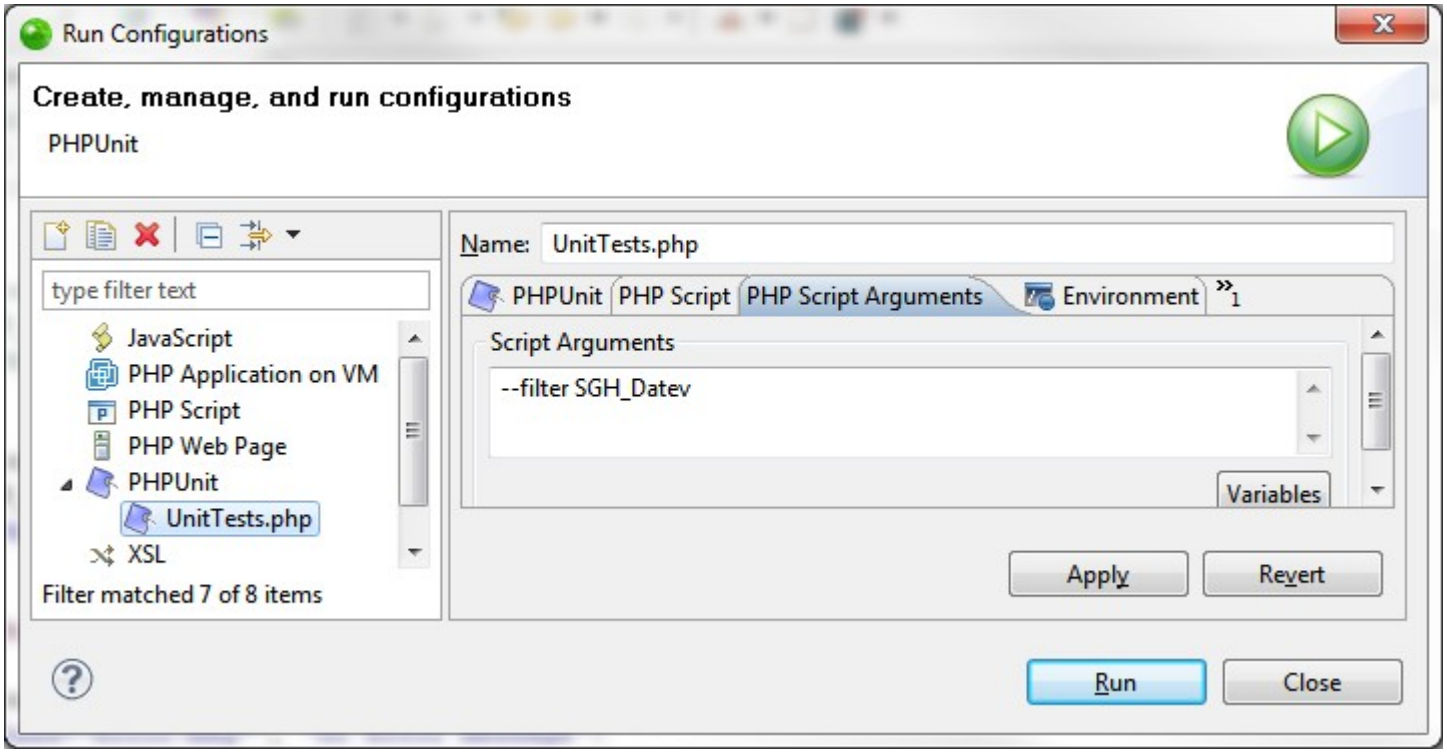
Quelle: EcomDev_PHPUnit Manual

Test-Durchführung

- Kommandozeile

```
path/to/magento$ phpunit --filter Vendor_Module UnitTests.php
```

- Zend Studio (und andere IDEs analog)



Unit Tests für JavaScript

- Verschiedene konkurrierende Frameworks
- In dieser Präsentation:
  - JSTestDriver
  - Qunit
- Testgetriebene Entwicklung in JS
  - Unterstützt saubereren, modularen Aufbau
  - Sinnvoll für nicht-trivialen Code

JSTestDriver

http://code.google.com/p/js-test-driver/

- xUnit Test Framework
- IDE Integration (Eclipse, PhpStorm)
- Läuft ohne Browser
- **Achtung:** Inkompatibel mit prototype.js
  - Inoffizieller Fix: http://goo.gl/bQXMh

QUnit

http://docs.jquery.com/Qunit

- Test Framework von (aber nicht nur für) jQuery
- Läuft in Browser (HTML-Dokument)
  - Fixture und Test Runner zugleich
- Aber auch als Plugin für JsTestDriver verfügbar

  → Automatisierung

  → Einschränkung: Kein Zugriff auf DOM Fixture

QUnit Test

QUnit example ☐ noglobals ☐ notrycat

☐ Hide passed tests

Mozilla/5.0 (Windows NT 6.1; WOW64; rv:9.0.1) Gecko/20100101

Tests completed in 66 milliseconds.
12 tests of 12 passed, 0 failed.

1. productvideos: onFileQueued: progressbar created (0, 7, 7) Rerun
2. productvideos: onFileQueueError (0, 1, 1
3. productvideos: clickCancel (0, 2, 2) Reru
   1. cancel button created
   2. upload canceled on click
4. productvideos: onUploadStart (0, 1, 1)
5. productvideos: onUploadError (0, 1, 1)

```
module("productvideos", {
    setup : function()
    {
        this.up = new SGH.Upload();
        $("sgh_productvideos_progress").innerHTML = "";
    },
    teardown : function()
    {
        this.up = null;
    },
```

Fixture

```
test("clickCancel", function() {
    this.up.onFileQueued({ index: 0, id: 1, name: "foo.mp4"});
    var row = $('sgh_productvideos_progress').childElements()[0];

    this.up.drawer.createCancelButton(
            {index: 0, id: 1}, this.up, row.childElements()[2]
    );
    var button = row.childElements()[2].childElements()[1];
    equal(button.innerHTML, "Cancel", "cancel button created");
    button.simulate("click");
    equal(this.up.uploader.canceled, 1, "upload canceled on click");
});
```

Assertions

Simulation von Interaktion