Functional Testing with Symfony2

Symfony Live 2013, Portland

Benjamin Eberlei
Qafoo GmbH
23th May 2013

Me

- Working at Qafoo

Helping people to create high quality web applications.

`http://qafoo.com`

- Doctrine Developer
- Symfony Contributor (usually some days before feature freeze)
- Twitter `@beberlei` and `@qafoo`

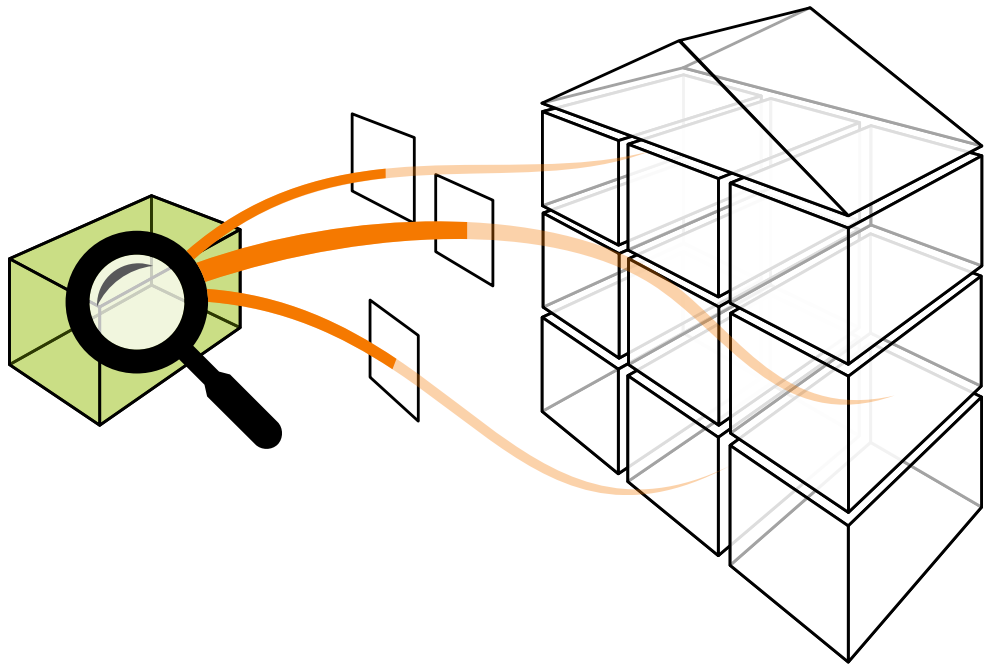
Copyright Qafoo GmbH; All Rights Reserved

Testing in Symfony

- Unit-Tests
- Integration/Functional Tests
- Acceptance Tests
- System Tests (End-To-End)

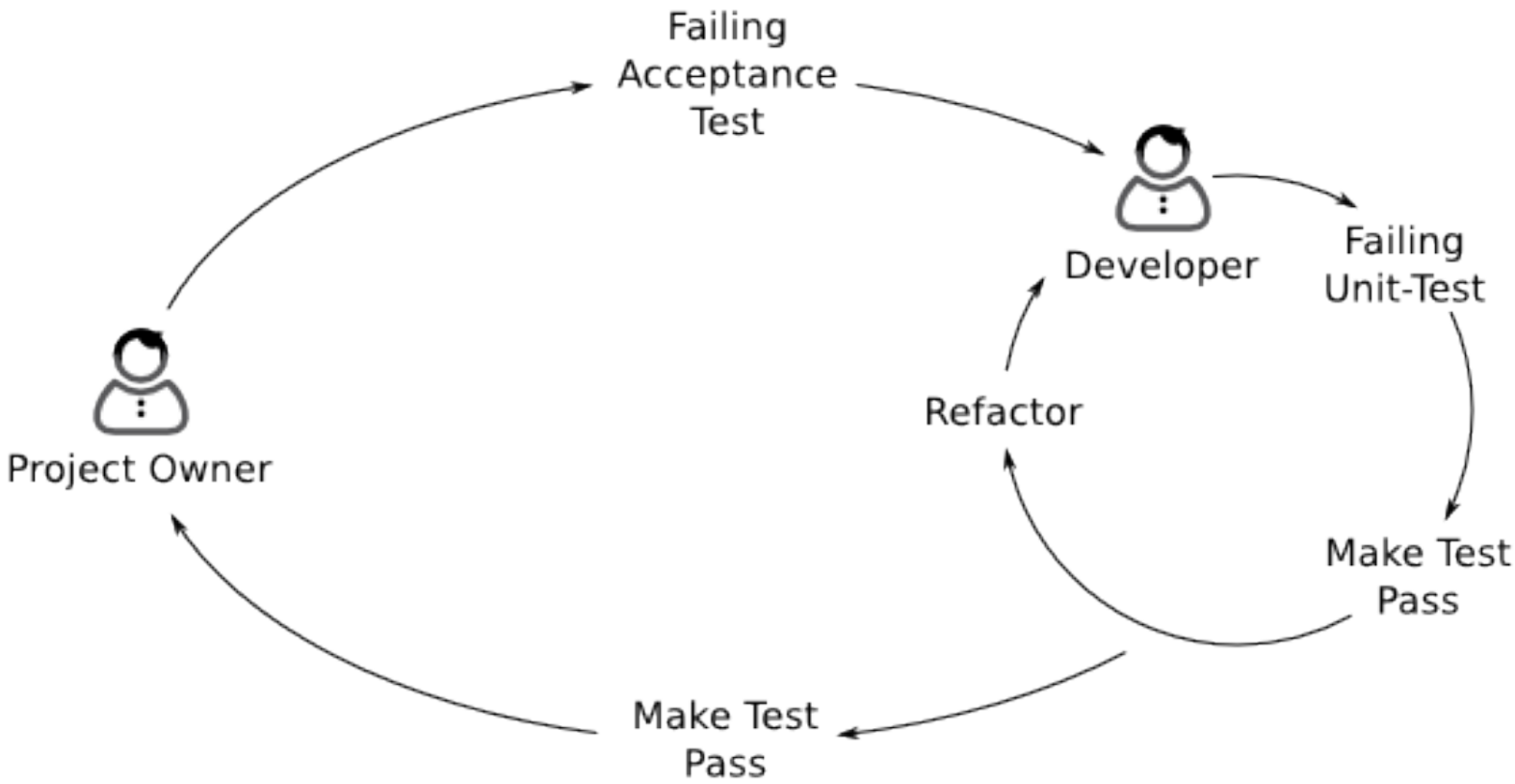
Copyright Qafoo GmbH; All Rights Reserved

Unit Testing



Copyright Qafoo GmbH; All Rights Reserved

Use Multiple Feedback Cycles



Copyright Qafoo GmbH; All Rights Reserved

Use Multiple Feedback Cycles

- Long cycles, slow tests
- Short cycles, fast tests
- Find a good mix between them

Copyright Qafoo GmbH; All Rights Reserved

Symfony Functional Tests

- Application as HTTP blackbox
- Use a Browser (Client) for interaction
- Assertions on HTTP responses
- `WebTestCase` integration into PHPUnit
- Symfony2 Extension for Behat

Copyright Qafoo GmbH; All Rights Reserved

Example: WebTestCase

```php
<?php
namespace WeatherBundle\Tests\Controller;

use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;

class DetailsControllerTest extends WebTestCase
{
    public function testShowGelsenkirchenWeather()
    {
        $client = static::createClient();

        $crawler = $client->request('GET', '/weather/zip/45887');
        $lastContent = $client->getResponse()->getContent();

        $this->assertRegex(
            '(([0-9]{2}) Grad Celsius)',
            $lastContent
        );
        $this->assertContains('Gelsenkirchen', $lastContent);
    }
}
```


Copyright Qafoo GmbH; All Rights Reserved

phpunit.xml

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<phpunit>
    <!-- .. -->
    <php>
        <server name="KERNEL_DIR" value="app/" />
    </php>

    <testsuites>
        <testsuite name="MyProject Testsuite">
            <directory>src/*/*Bundle/Tests</directory>
            <directory>src/*/Bundle/*Bundle/Tests</directory>
        </testsuite>
    </testsuites>
    <!-- .. -->
</phpunit>
```


Copyright Qafoo GmbH; All Rights Reserved

Symfony Component: BrowserKit

- `Symfony\Component\BrowserKit\Client` object
- API to simulate a browser session
- Handles Cookies, Session and History
- Symfony HttpKernel implementation
  - wrapped around your application kernel
  - Stimulate `HttpKernelInterface::handle`
  - `@igorw`'s session "The HttpKernelInterface is a lie"
- Goutte
  - Testing "over the wire" using Guzzle HTTP client
  - Time-Travel back to `@mtdowling`'s session yesterday

Copyright Qafoo GmbH; All Rights Reserved

Client#request()

Parameter	Description
`$method`	`GET`, `POST`, ...
`$uri`	Relative Uri to Base Path with Query String
`$parameters`	POST Parameters
`$files`	Uploaded Files
`$server`	Headers and Environment
`$content`	POST Body String

Copyright Qafoo GmbH; All Rights Reserved

Symfony Component: DomCrawler

- Simple API for HTML/XML traversal
- Similar to jQuery API
- Two traversal languages
  - XPath with `filterXpath($xPathSelector)`
  - CSS Selector with `filter($cssSelector)`
- `Client::request()` returns a `Crawler` instance

Copyright Qafoo GmbH; All Rights Reserved

Filter XPath Example

```php
<?php
namespace WeatherBundle\Tests\Controller;

use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;

class DetailsControllerTest extends WebTestCase
{
    public function testShowTemperature()
    {
        $client = static::createClient();

        $crawler = $client->request('GET', '/weather/zip/45887');

        $this->assertRegex(
            '(([0-9]{2}) Grad Celsius)',
            $crawler->filterXPath('//span[@id="temperature"]')->text()
        );
    }
}
```


Copyright Qafoo GmbH; All Rights Reserved

Filter CSS Example

```php
<?php
namespace WeatherBundle\Tests\Controller;

use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;

class DetailsControllerTest extends WebTestCase
{
    public function testShowTemperature()
    {
        $client = static::createClient();

        $crawler = $client->request('GET', '/weather/zip/45887');

        $this->assertRegex(
            '(([0-9]{2}) Grad Celsius)',
            $crawler->filter('span#temperature')->text()
        );
    }
}
```


Copyright Qafoo GmbH; All Rights Reserved

DomCrawler Node API

```php
<?php

$crawler = $client->request('GET', '/weather/53225');

$this->assertEquals('Weather: Bonn', $crawler->filter('title')->text());
$this->assertEquals('53225', $crawler->filter('#loc')->attr('data-zip'));

$this->assertEquals('First', $crawler->filter('td')->first()->text());
$this->assertEquals('Last', $crawler->filter('td')->last()->text());
$this->assertEquals('10th', $crawler->filter('td')->eq(10)->text());

$this->assertEquals(10, $crawler->filter('ul')->children()->count());

$mainElement = $crawler->filterXPath('//div[@id="main"]');
$this->assertEquals(array("data-id" => "1234"), $mainElement->extract(array("data-id")));
```


Copyright Qafoo GmbH; All Rights Reserved

Click Links and Submit Forms

- Use the Client like a real browser
- `Crawler` and `Client` interaction
- 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

Copyright Qafoo GmbH; All Rights Reserved

Client: Click Links

```php
<?php
namespace WeatherBundle\Tests\Controller;

use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;

class DetailsControllerTest extends WebTestCase
{
    public function testNextDayGelsenkirchenWeather()
    {
        $client = static::createClient();

        $crawler = $client->request('GET', '/weather/zip/45887');
        $link = $crawler->selectLink('Next Day')->link();

        $tomorrowCrawler = $client->click($link);

        $lastContent = $client->getResponse()->getContent();
        $this->assertContains('Tomorrow', $lastContent);
    }
}
```


Copyright Qafoo GmbH; All Rights Reserved

Client: Submit Forms

```php
<?php
namespace WeatherBundle\Tests\Controller;

use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;

class DetailsControllerTest extends WebTestCase
{
    public function testNextDayGelsenkirchenWeather()
    {
        $client = static::createClient();

        $crawler = $client->request('GET', '/weather/zip/45887');
        $form = $crawler->selectButton('Send to Friends')->form();

        $confirmCrawler = $client->submit($form, array(
            'send[email]' => 'toby@qafoo.com',
        ));

        $client->followRedirect();

        $lastContent = $client->getResponse()->getContent();
        $this->assertContains('Weather sent to toby@qafoo.com', $lastContent);
    }
}
```


Copyright Qafoo GmbH; All Rights Reserved

Test-Setup Considerations

- Shared Fixture vs Isolated Fixture
- Database Setup
- Security

Copyright Qafoo GmbH; All Rights Reserved

Shared vs Isolated Fixture

- Shared Fixture
  - All tests read/write on the same data
  - No isolation of the tests
  - Requires many datasets/rows
- Isolated Fixture
  - Every test uses its own data
  - Isolation of tests
  - Slow test-setup
  - Micromanagement of fixtures

Copyright Qafoo GmbH; All Rights Reserved

Database Setup

- When to create the database schema?
  - CI requires automation of schema creation
  - Creating a database is very expensive (I/O)
- Use SQLite or the ”real” database?
  - SQLite can run ”in memory”
  - But is SQLite compatible with your application?
- Use shared database conncetion

Copyright Qafoo GmbH; All Rights Reserved

How to load fixtures?

- Different solutions
  - SQL dumps
  - PHPUnit DBUnit XMLs
  - Doctrine Fixtures
- Keep it simple

Copyright Qafoo GmbH; All Rights Reserved

How to load fixtures?

```php
<?php
namespace WeatherBundle\Tests\Controller;

use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;

abstract class ProjectWebTestCase extends WebTestCase
{
    static protected function createKernel(array $options = array())
    {
        $kernel = parent::createKernel($options);

        // fixture setup here

        return $kernel;
    }
}
```


Copyright Qafoo GmbH; All Rights Reserved

Shared Database Connection

```php
<?php
namespace WeatherBundle\Tests\Controller;

use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;

abstract class ProjectWebTestCase extends WebTestCase
{
    static protected $sharedDatabase;

    static protected function createKernel(array $options = array())
    {
        $kernel = parent::createKernel($options);

        if (self::$sharedDatabase === null) {
            self::$sharedDatabase = $kernel->getContainer()
                ->get('doctrine.dbal.default_connection');
        }

        $kernel->getContainer()->set(
            'doctrine.dbal.default_connection', self::$sharedDatabase);

        // load fixtures here

        return $kernel;
    }
}
```


Copyright Qafoo GmbH; All Rights Reserved

Golden Rule for Database Testing

Don't use `tearDown()` to reset fixtures

Copyright Qafoo GmbH; All Rights Reserved

Security Setup Options

1. Use real authentication
2. Disable authentication
3. Change authentication strategy

Copyright Qafoo GmbH; All Rights Reserved

Security Setup

```
# app/config/config_test.yml
security:
    firewalls:
        main:
            pattern: ^/
            http_basic: ~
```

```
<?php
namespace WeatherBundle\Tests\Controller;

use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;

class AdminControllerTest extends WebTestCase
{
    public function testPerformAdminOperation()
    {
        $client = static::createClient(array(), array(
            'PHP_AUTH_USER' => 'admin',
            'PHP_AUTH_PW'   => 's3cr3t',
        ));
        // ...
    }
}
```


Copyright Qafoo GmbH; All Rights Reserved

Behat

- Behavior Driven Development
- Gherkin Language to describe acceptance criteria
- Mink: A Gherkin language for browser Interaction
- Symfony2 Extension using BrowserKit

Copyright Qafoo GmbH; All Rights Reserved

Behat: Example Revisted

```
1  Feature: Show Weather
2      In order to see the weather of my city
3      As a user of the website
4      I need to search and view the weather
5
6      Scenario: Show Weather by ZIP code
7          Given I am on "/weather"
8          When I fill "weather[zip]" with "45887"
9          And I press "Show Weather"
10         Then I should be on "/weather/zip/45887"
11         And I should see "([0-9]{2}) Grad Celsius)"
12         And I should see "Gelsenkirchen"
```


Copyright Qafoo GmbH; All Rights Reserved

Behatify PHPUnit WebTestCase

```php
<?php
namespace WeatherBundle\Tests\Controller;

use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;

class DetailsControllerTest extends WebTestCase
{
    private $client;
    private $crawler;

    public function setUp()
    {
        $this->client = static::createClient();
    }

    public function givenIAmOn($url)
    {
        $this->crawler = $client->request('GET', $url);
    }

    public function whenISubmit($button, array $formValues)
    {
        $form = $crawler->selectButton($button)->form();
        $this->crawler = $client->submit($form, $formValues);
    }

    // ...
}
```

Qafoo
passion for software quality
Copyright Qafoo GmbH; All Rights Reserved

Behatify PHPUnit WebTestCase

```php
<?php
namespace WeatherBundle\Tests\Controller;

use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;

class DetailsControllerTest extends WebTestCase
{
    private $client;
    private $crawler;

    // ...

    public function thenIShouldSee($text)
    {
        $lastContent = $this->client->getResponse()->getContent();
        $this->assertContains($text, $lastContent);
    }

    public function testShowGelsenkirchenWeather()
    {
        $this->givenIAmOn('/weather');
        $this->whenISubmit('Show weather', array(
            'weather[zip]' => '45887'
        ));
        $this->thenIShouldSee('Gelsenkirchen');
    }
}
```


Copyright Qafoo GmbH; All Rights Reserved

Integration Tests with Container

- Using Third-Party Code requires Integration tests
  - Don't trust their API
  - Caution when mocks simulate wrong behavior
- Examples:
  - Code relies on `INSERT +ORDER BY` of database
  - Webservice breaks format or has unreliable uptime
  - Symfony `Container` is case-insensitive on service ids
- Solution: Use real Container to construct third party dependencies
  - Use Symfony `test` environment

Copyright Qafoo GmbH; All Rights Reserved

Integration Tests with Container

```php
<?php

use Symfony\Bundle\FrameworkBundle\Test\WebTestCase;

abstract class KernelTestCase extends WebTestCase
{
    protected $container;

    public function setUp()
    {
        $config = array('environment' => 'test', 'debug' => true);
        self::$kernel = self::createKernel($config);
        self::$kernel->boot();

        $this->container = self::$kernel->getContainer();
    }

    public function tearDown()
    {
        if (self::$kernel === null) {
            return;
        }

        self::$kernel->shutdown();
    }
}
```


Copyright Qafoo GmbH; All Rights Reserved

Integration Tests with Container

```php
<?php

class MyServiceTest extends KernelTestCase
{
    private $service;

    public function setUp()
    {
        $entityManager = $this->container->get('doctrine.orm.default_entity_manager');
        $mailer = $this->container->get('mailer');

        $this->service = new MyService($entityManager, $mailer);
    }

    public function testSomething()
    {
        $this->service->something();
    }
}
```


Copyright Qafoo GmbH; All Rights Reserved

Tip: Test Construction of all your Services

```php
<?php

class MyBundleContainerTest extends KernelTestCase
{
    /**
     * @dataProvider dataServiceConstruction
     */
    public function testServiceConstruction($id, $class)
    {
        $service = $this->container->get($id);
        $this->assertInstanceOf($class, $service);
    }

    static public function dataServiceConstruction()
    {
        return array(
            array('acme_demo.my_service', 'Acme\DemoBundle\Service\MyService'),
            array('acme_demo.repository.user', 'Acme\DemoBundle\Entity\UserRepository'),
        );
    }
}
```


Copyright Qafoo GmbH; All Rights Reserved

Questions?

Stay in touch

- Benjamin Eberlei
- benjamin@qafoo.com
- @beberlei / @qafoo

Copyright Qafoo GmbH; All Rights Reserved

THANK YOU

Rent a quality expert
qafoo.com

Copyright Qafoo GmbH; All Rights Reserved