



A Sweet Test Suite

DrupalCamp Atlanta | A Sweet Test Suite | Dan Gurin | @dgurin | @CIVICACTIONS

CivicActions

A Sweet Test Sweet

Dan Gurin

Engineer @CivicActions

Drupal Camp Asheville Organizer

Twitter @dgurin

**dangur on github, LinkedIn, www,
Slack...**



Test Driven Development

1. Behavior-Driven Development (BDD) with Behat

2. Visual Regression Testing

3. Unit testing with PHPUnit



Behat Drupal Extension

- Plain language
- Common scenarios
- Changes & updates
- Extensible extension



Feature: Some terse yet descriptive text of what is desired

In order to realize a named business value

As an explicit system actor

I want to gain some beneficial outcome which furthers the goal

Scenario: Some determinable business situation

Given some precondition

And some other precondition

When some action by the actor

And some other action

And yet another action

Then some testable outcome is achieved

And something else we can check happens too

Scenario: A different situation

DrupalCamp Atlanta | A Sweet Test Suite | Dan Gurin | @dgurin | @CIVICACTIONS

...

CivicActions

Tags

- **Driver @api @javascript**
- **Groups @database @form**
- **Hooks**
 - @beforeNodeCreate
 - @AfterScenario

Drivers

Blackbox		No privileges
Drush	@api	Can run on a different server
Drupal	@api	More step definitions

machine_name *

Machine Name

**Generated by JavaScript, so
without PhantomJS & @javascript,
becomes an additional required
field.**

Comments

- **# <- Anything on a line followed by a hashtag**
 - **Use PhpStorm?**
- Control or Command Slash /
to comment out line(s)**

Background & Tables

- Applies to all Scenarios
- Create many items

Background:

Given pages:

name	url	
Groups	/groups	
Content	/admin/content/	

Given users:

name	mail	roles	
John	john@example.com	site manager	
Badmin	admin@example.com	site manager	
Gabriel	gabriel@example.com	editor	
Jaz	jaz@example.com	editor	
Katie	katie@example.com	editor	
Martin	martin@example.com	editor	
Celeste	celeste@example.com	editor	

Given groups:

title	author	published	
Group 01	Badmin	Yes	
Group 02	Badmin	Yes	
Group 03	Badmin	No	

And group memberships:

user	group	role on group	membership status	
Gabriel	Group 01	administrator member	Active	
Katie	Group 01	member	Active	
Jaz	Group 01	member	Pending	
Celeste	Group 02	member	Active	

Contexts

- **Drupal - users, terms, and nodes**
 - DrupalRaw - functionality
- **Mink - regions and forms**
- **Markup - tags, classes, & attributes**
- **Message - notice, warning, & error**
- **Drush - call commands directly**

Custom Contexts

FeatureContext.php file

CustomContext class

- 1. Write the step**
- 2. Run the test (it will fail)**
- 3. Copy the generated snippet**

Custom Contexts

```
--- Drupal\FeatureContext has missing steps. Define them with these snippets:

/**
 * @Given I am doing something new and exciting
 */
public function iAmDoingSomethingNewAndExciting()
{
    throw new PendingException();
}
```



Custom Contexts

```
PHP Fatal error:  Class 'Drupal\PendingException' not  
Fatal error: Class 'Drupal\PendingException' not found  
[Testing\Behat] Exit code 255  
[error] Behat tests failed!
```


Custom Contexts

- 1. Delete exception**
- 2. Add code**
- 3. Test again and see if it worked**

Debugging

- **Steps**
 - print current url
 - print last response
- **`print_r($variable);`**

D8 extension extension

github.com/nuvoleweb/drupal-behat

**Autocomplete, CKEditor, Chosen,
Content, Drupal, Menu, Position,
Raw, Responsive, Screenshot,
Select, ServiceContainer, Taxonomy,
Visibility**

Sub contexts

- **Contrib & Custom projects**
 - **Modules**
 - **Themes**

Contributing

- Behat Drupal Extension
- <https://github.com/Behat/Behat/blob/master/CONTRIBUTING.md>
- <http://mink.behat.org/en/latest/contributing.html>

Test Driven Development

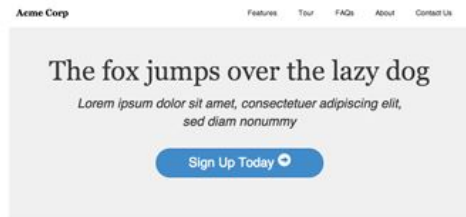
1. Behavior-Driven Development (BDD) with Behat
- 2. Visual Regression Testing**
3. Unit testing with PHPUnit



Visual Regression Testing coverage

- Display changes
- Great for updates

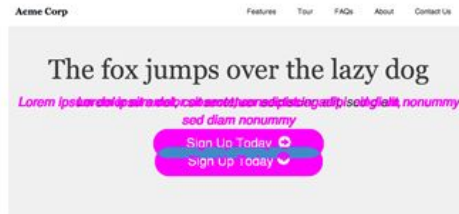
Visual Regression Test



after

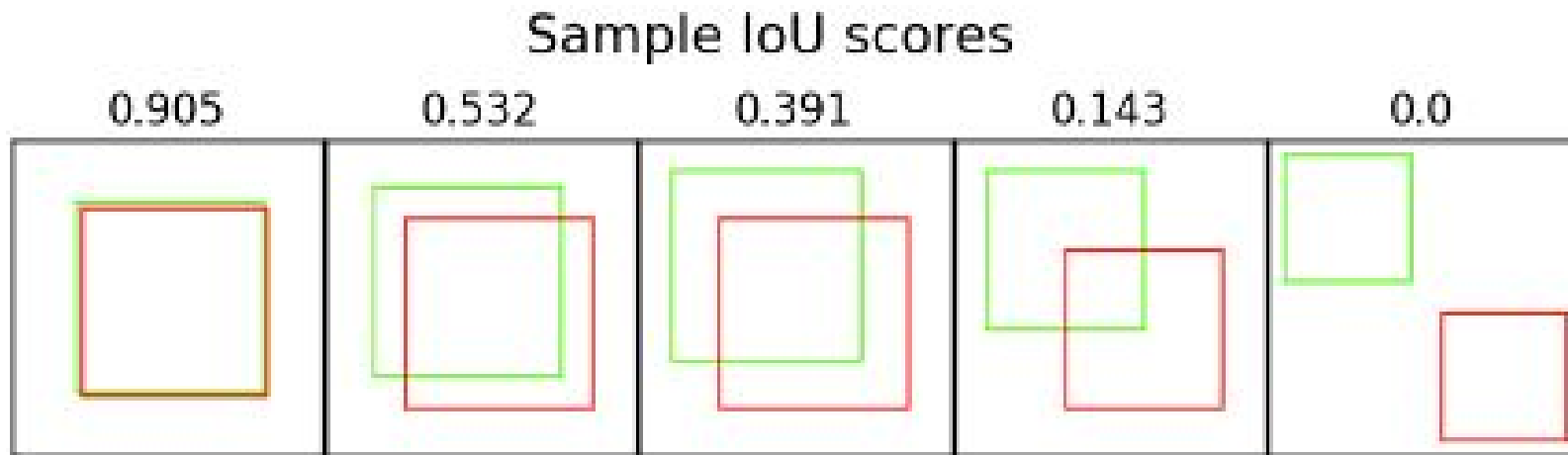


before



diff

Intersection over Union IoU



Test Driven Development

1. Behavior-Driven Development (BDD) with Behat
2. Visual Regression Testing
3. Unit testing with PHPUnit



Drupal 8 PHPUnit Suites

<u>Name</u>	<u>Extend</u>	<u>State</u>	<u>Requires</u>
Unit	UnitTestCase	None	Code
Kernel	KernelTestBase	Bootstrapped	Database
Functional	BrowserTestBase	Booted	Webserver
Javascript	JavascriptTestBase	Booted	PhantomJS

Unit

```
use Drupal\Tests\UnitTestCase;
class TheClassImTesting extends UnitTestCase {
    public function testExample() {
        $this->assertEquals( expected: 1, actual: 1);
    }
}
```

Assertions

- Equals
- Same
- True / False
- **PHPUnit Documentation Appendix A**

<https://phpunit.de/manual/current/en/phpunit-book.html#appendixes.assertions>

Kernel

- Setup
- Config



Setup

- **Beginning of class**
- **Applies to all test functions**
- **Set up the parent first**

Install

- **Schema**
- **Entity schema**
- **Configuration**

setUp

```
use Drupal\KernelTests\KernelTestBase;
class TheClassImKernelTesting extends KernelTestBase {
    protected function setUp() {
        parent::setUp();
        $this->installSchema( module: 'webform', [
            'webform_submission',
            'webform_submission_data',
        ]);
        $this->installEntitySchema( entity_type_id: 'node');
        $this->installConfig(['system', 'maillog']);
    }
}
```

Install Fields

- Get from configuration management
- Create
 - Storage
 - Field

```
$yaml = yaml_parse(file_get_contents( filename: $configPath . "/field.storage.{entityType}.{fieldName}.yaml"));  
FieldStorageConfig::create($yaml)->save();  
$yaml = yaml_parse(file_get_contents( filename: $configPath . "/field.field.{entityType}.{bundle}.{fieldName}.yaml"));  
FieldConfig::create($yaml)->save();
```

Enable modules

```
/**  
 * {@inheritdoc}  
 */  
public static $modules = ['webform'];
```

Functional

- Enable modules
- Paths!
- Web assert

Functional

```
use Drupal\Tests\BrowserTestBase;
class MyClassTest extends BrowserTestBase {
    public function testMyTest() {
        $this->drupalGet( path: '' ); // Gets homepage.
        $this->assertSession()->statusCodeEquals(200);
        $this->assertSession()->linkExists('Blog');
    }
}
```

JavaScript

- **executeScript / evaluateScript**
- **switchToFrame**
- **keyDown**
- **resizeWindow**
- **wait**

JavascriptTestBase

- **createScreenshot**
- **getDrupalSettings**

JavaScript Assertions

- **assertElement(Not)Visible**
- **assertJsCondition**

assertSession

- **waitFor**
AjaxToFinish, Button, Element(Visible),
Field, Link
- **waitOnAutocomplete**
- **responseContains**
- **isVisible**

Test Doubles

**Return a value
without the object**



Mock Objects

```
$webform = $this->getMockBuilder( className: 'Drupal\webform\Entity\Webform')
    ->disableOriginalConstructor()
    ->setMethods(['id'])
    ->getMock();
$webform->expects($this->once())
    ->method('id')
    ->will($this->returnValue( value: 'a_test_webform'));

$webformId = $webform->id();
```

Traits

- **Code reuse in many classes**
- **Groups sets of methods**
- **Can not be instantiated**

Reflection Trait

Test protected / private methods

```
public function invokeMethod(&$object, $methodName, array $parameters = []) {  
    $reflection = new \ReflectionClass(get_class($object));  
    $method = $reflection->getMethod($methodName);  
    $method->setAccessible(TRUE);  
  
    return $method->invokeArgs($object, $parameters);  
}
```

Trait in action

```
class  extends KernelTestBase {
```

```
    use ReflectionTrait;
```

```
    $assembledData = $this->invokeMethod($this->submissionServiceApi, 'assembleRequestData', [$webformSubmission, $webform]);
```

Data Providers

- Run same test with different datasets

```
/**
 * @dataProvider sumProvider
 */
public function testSum($a, $b, $sum) {
    $this->assertEquals($sum, $a + $b);
}

public function sumProvider() {
    return [
        [1, 2, 3],
        [4, 5, 9],
        [7, 8, 9]
    ];
}
```

Data Providers

- Named dataset via associative array

```
/**
 * @dataProvider sumProvider
 */
public function testSum($a, $b, $sum) {
    $this->assertEquals($sum, $a + $b);
}

public function sumProvider() {
    return [
        'easy as' => [1, 2, 3],
        'niner' => [4, 5, 9],
        'false' => [7, 8, 9]
    ];
}
```


Debugging

- **`print_r($whatEver);`**
- **Screenshots**
- **Reports**
- **Error/failure messages**

Running tests

- **phpunit with arguments**
- **blt:tests phpunit**
- **CI**
- **@group**
- **. = pass, E = error, F = Failure**

Happy & Sad Paths

- Test failure as well as success
- **expectException()**
 - Code, Message, MessageRegExp



Base Classes

- **Common setups**
- **Nested levels**

Open Discussion

Thank you.