

Testing ../modules/override\_node\_options

....

Time: 2.54 minutes, Memory: 6.00MB

OK (4 tests, 65 assertions)

docker@cli:/var/www/core\$

# TDD: TEST

# DRIVEN DRUPAL

**OPDAVI.ES/TDD-TEST-DRIVEN-DRUPAL**

**DIAMOND SPONSOR**

AcQUID<sup>®</sup>

# PLATINUM SPONSORS





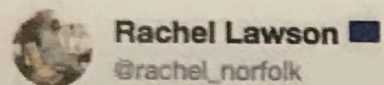
# GOLD SPONSORS








## Drupal Association Signup



Rachel Lawson   
@rachel\_norfolk

As I declared in the questions/answers after my talk @drupaldevdays, I challenge you all...

If 200 Dev Days participants sign up to be Drupal Association members this week, I will post a video of Megan's reaction. [drupal.org/association/in...](http://drupal.org/association/in...) - it's less than a Euro a week!



*bit.ly / devdays - join - da*



- ▶ **Module and theme developers**
- ▶ **Want to know more about automated testing**
- ▶ **Looking to start writing your first tests**
- ▶ **Drupal 8**
- ▶ **PHPUnit**

- ▶ **Why write tests, and what to test**
- ▶ **Types of tests**
- ▶ **How to run tests**
- ▶ **Real life example**
- ▶ **Building a new module with TDD**

- ▶ **Full Stack Web Developer & System Administrator**
- ▶ **Senior Developer at Microserve**
- ▶ **Part-time freelancer**
- ▶ **Acquia certified Drupal 8 Grand Master**
- ▶ **Drupal 7 & 8 core contributor**
- ▶ **Symfony, Laravel, ~~Silex~~, Sculpin**
- ▶ **@opdavies**
- ▶ **[www.oliverdavies.uk](http://www.oliverdavies.uk)**



- ▶ [oliverdavies.uk/talks](https://oliverdavies.uk/talks)
- ▶ [oliverdavies.uk/twitter](https://oliverdavies.uk/twitter)
- ▶ [oliverdavies.uk/drupal](https://oliverdavies.uk/drupal)
- ▶ [oliverdavies.uk/github](https://oliverdavies.uk/github)

- ▶ <https://microserve.io>
- ▶ <https://www.drupal.org/microserve>
- ▶ <https://github.com/microserve-io>
- ▶ <https://twitter.com/microserveltd>
- ▶ <https://www.linkedin.com/company/microserve-ltd>



**TESTDRIVENDRUPAL.COM**



**WRITE CUSTOM MODULES  
AND THEMES FOR CLIENTS**

**OCCASSIONALLY  
CONTRIBUTE TO CORE**

**MAINTAIN AND CONTRIBUTE  
TO CONTRIB PROJECTS**



**Tim Millwood** 

@timmillwood

Following



Anyone wanna become maintainer or co-maintain Override Node Options [#Drupal](#) module?

6:29 PM - 23 Feb 2012

# OVERRIDE NODE OPTIONS

- ▶ Become maintainer in 2012
- ▶ Had some existing tests
- ▶ Used on **11,046 sites** in October 2012 (**84 D5, 7,094 D6, 3,868 D7**)
- ▶ Used on **29,023 sites** in June 2018 (**9 D5, 1,853 D6, 23,602 D7, 3,559 D8**)
- ▶ **#236** most used module on Drupal.org
- ▶ Crucial to preventing regressions when adding new features or fixing bugs

**WHY WRITE TESTS?**

# WHY WRITE TESTS?

- ▶ Catch bugs earlier
- ▶ Piece of mind
- ▶ Prevent regressions
- ▶ Write less code
- ▶ Documentation
- ▶ Drupal core requirement - <https://www.drupal.org/core/gates#testing>
- ▶ More important with regular D8 releases

# CORE TESTING GATE

- ▶ **New features should be accompanied by automated tests.**
- ▶ **If the feature does not have an implementation, provide a test implementation.**
- ▶ **Bug fixes should be accompanied by changes to a test (either modifying an existing test case or adding a new one) that demonstrate the bug.**



# TESTING IN DRUPAL

- ▶ **Drupal 7** - SimpleTest (testing) module provided as part of core
- ▶ **Drupal 8** - PHPUnit added as a core dependency
- ▶ **PHPUnit Initiative** - SimpleTest to be deprecated and removed in Drupal 9

# WRITING TESTS (DRUPAL 8)

- ▶ PHP class with `.php` extension
- ▶ `tests/src` directory within each module
- ▶ Within the `Drupal\Tests\module_name` namespace
- ▶ Class name must match the filename
- ▶ Namespace must match the directory structure
- ▶ One test class per feature
- ▶ Each method must start with `test`

**1. ARRANGE**

**2. ACT**

**3. ASSERT**

```
// modules/example/tests/src/Functional/ExampleTest.php
```

```
namespace Drupal\Tests\example\Functional;
```

```
use Drupal\Tests\BrowserTestBase;
```

```
class ExampleTest extends BrowserTestBase {
```

```
    public function testSomething() {
```

```
        // Arrange
```

```
        // Act
```

```
        // Assert
```

```
    }
```

```
}
```

# WHAT TO TEST?

- ▶ **Creating nodes with data from an API**
- ▶ **Calculating attendance figures for an event**
- ▶ **Determining if an event is purchasable**
- ▶ **Promotions and coupons for new users**
- ▶ **Cloning events**
- ▶ **Queuing private message requests**
- ▶ **Emails for new memberships**
- ▶ **Closed support tickets are re-opened when comments are added**
- ▶ **Custom form validation rules**

# WHAT TO TEST FIRST?

- ▶ **What is the core piece of functionality?**
- ▶ **What provides the most value to the client?**
- ▶ **What would you not like to be fixing on a Friday afternoon or after hours?**



**Povilas@LaravelDaily**

@DailyLaravel

Following



No.1 automatic test you should write is for the function that would make you lose your job, if it breaks. [#quoteOfTheDay](#) by [@stauffermatt](#) at [@LaravelLiveUK](#)

10:33 AM - 12 Jun 2018

# WHAT TO TEST FIRST?

- ▶ Write a **new test** when adding any **new functionality**
- ▶ Write a **regression test** when **fixing a bug**



# TYPES OF TESTS

- ▶ **Unit**
- ▶ **Kernel (integration)**
- ▶ **Functional / FunctionalJavascript (web, browser, feature)**

# UNIT TESTS

- ▶ **Tests PHP logic**
- ▶ **No database interaction**
- ▶ **Fast to run**
- ▶ **Need to mock dependencies**
- ▶ **Can become tightly coupled**
- ▶ **Can be hard to refactor**

```
// tests/src/Unit/JobTest.php

namespace Drupal\Tests\advancedqueue\Unit;

use Drupal\advancedqueue\Job;
use Drupal\Tests\UnitTestCase;

class JobTest extends UnitTestCase {

    public function testCreate() {
        $job = Job::create('test', ['my' => 'data']);

        $this->assertEquals('test', $job->getType());
        $this->assertEquals(['my' => 'data'], $job->getPayload());
        $this->assertEquals(Job::STATE_QUEUED, $job->getState());
    }
}
```

```
// tests/src/Unit/JobTest.php

namespace Drupal\Tests\advancedqueue\Unit;

use Drupal\advancedqueue\Job;
use Drupal\Tests\UnitTestCase;

class JobTest extends UnitTestCase {

    public function testCreate() {
        $job = Job::create('test', ['my' => 'data']);

        $this->assertEquals('test', $job->getType());
        $this->assertEquals(['my' => 'data'], $job->getPayload());
        $this->assertEquals(Job::STATE_QUEUED, $job->getState());
    }

}
```

```
// tests/src/Unit/JobTest.php

namespace Drupal\Tests\advancedqueue\Unit;

use Drupal\advancedqueue\Job;
use Drupal\Tests\UnitTestCase;

class JobTest extends UnitTestCase {

    public function testCreate() {
        $job = Job::create('test', ['my' => 'data']);

        $this->assertEquals('test', $job->getType());
        $this->assertEquals(['my' => 'data'], $job->getPayload());
        $this->assertEquals(Job::STATE_QUEUED, $job->getState());
    }

}
```

# KERNEL TESTS

- ▶ **Integration tests**
- ▶ **Can install modules, interact with services, container, database**
- ▶ **Minimal Drupal bootstrap**
- ▶ **Faster than functional tests**
- ▶ **More setup required**

```
// tests/src/Kernel/ProcessorTest.php

namespace Drupal\Tests\advancedqueue\Kernel;

use Drupal\advancedqueue\Entity\Queue;
use Drupal\advancedqueue\Job;
use Drupal\KernelTests\KernelTestBase;

class ProcessorTest extends KernelTestBase {

    ...

}
```

```
// tests/src/Kernel/ProcessorTest.php

protected function setUp() {
    parent::setUp();

    $this->installSchema('advancedqueue', ['advancedqueue']);

    $this->queue = Queue::create([
        'id' => 'test',
        'label' => 'Test queue',
        'backend' => 'database',
        'backend_configuration' => [
            'lease_time' => 5,
        ],
    ]);
    $this->queue->save();

    $this->processor = $this->container->get('advancedqueue.processor');
}
```



```
// tests/src/Kernel/ProcessorTest.php

protected function setUp() {
    parent::setUp();

    $this->installSchema('advancedqueue', ['advancedqueue']);

    $this->queue = Queue::create([
        'id' => 'test',
        'label' => 'Test queue',
        'backend' => 'database',
        'backend_configuration' => [
            'lease_time' => 5,
        ],
    ]);
    $this->queue->save();

    $this->processor = $this->container->get('advancedqueue.processor');
}
```

```
// tests/src/Kernel/ProcessorTest.php

protected function setUp() {
    parent::setUp();

    $this->installSchema('advancedqueue', ['advancedqueue']);

    $this->queue = Queue::create([
        'id' => 'test',
        'label' => 'Test queue',
        'backend' => 'database',
        'backend_configuration' => [
            'lease_time' => 5,
        ],
    ]);
    $this->queue->save();

    $this->processor = $this->container->get('advancedqueue.processor');
}
```

```
// tests/src/Kernel/ProcessorTest.php

protected function setUp() {
    parent::setUp();

    $this->installSchema('advancedqueue', ['advancedqueue']);

    $this->queue = Queue::create([
        'id' => 'test',
        'label' => 'Test queue',
        'backend' => 'database',
        'backend_configuration' => [
            'lease_time' => 5,
        ],
    ]);
    $this->queue->save();

    $this->processor = $this->container->get('advancedqueue.processor');
}
```

```
// tests/src/Kernel/ProcessorTest.php

public function testProcessor() {
    $first_job = Job::create('simple', [
        'test' => '1',
    ]);

    $second_job = Job::create('flexible', [
        'expected_state' => Job::STATE_SUCCESS,
        'expected_message' => 'Done!',
    ]);

    $third_job = Job::create(
        'flexible', ['expected_exception' => 'DB down!'],
    );

    $fourth_job = Job::create('flexible', [
        'expected_state' => Job::STATE_FAILURE,
        'expected_message' => 'Failed!',
    ]);

    ...
}
```

```
// tests/src/Kernel/ProcessorTest.php
```

```
public function testProcessor() {
```

```
    ...
```

```
    $this->queue->enqueueJob($first_job);
```

```
    $this->queue->enqueueJob($second_job);
```

```
    $this->queue->enqueueJob($third_job);
```

```
    $this->queue->enqueueJob($fourth_job);
```

```
    $num_processed = $this->processor->processQueue($this->queue);
```

```
    $this->assertEquals(4, $num_processed);
```

```
}
```

```
// tests/src/Kernel/ProcessorTest.php
```

```
public function testProcessor() {
```

```
    ...
```

```
    $this->queue->enqueueJob($first_job);
```

```
    $this->queue->enqueueJob($second_job);
```

```
    $this->queue->enqueueJob($third_job);
```

```
    $this->queue->enqueueJob($fourth_job);
```

```
    $num_processed = $this->processor->processQueue($this->queue);
```

```
    $this->assertEquals(4, $num_processed);
```

```
}
```

# FUNCTIONAL TESTS

- ▶ **Tests end-to-end functionality**
- ▶ **UI testing**
- ▶ **Interacts with database**
- ▶ **Full Drupal installation**
- ▶ **Slower to run**
- ▶ **With/without JavaScript**

```
// tests/src/Functional/QueueTest.php
```

```
namespace Drupal\Tests\advancedqueue\Functional;
```

```
use Drupal\advancedqueue\Entity\Queue;
```

```
use Drupal\advancedqueue\Entity\QueueInterface;
```

```
use Drupal\Tests\BrowserTestBase;
```

```
class QueueTest extends BrowserTestBase {
```

```
    ...
```

```
}
```



```
// tests/src/Functional/QueueTest.php
```

```
protected function setUp() {  
    parent::setUp();
```

```
    $this->placeBlock('local_tasks_block');
```

```
    $this->placeBlock('local_actions_block');
```

```
    $this->placeBlock('page_title_block');
```

```
    $this->adminUser = $this->drupalCreateUser(['administer advancedqueue']);
```

```
    $this->drupalLogin($this->adminUser);
```

```
}
```

```
// tests/src/Functional/QueueTest.php
```

```
protected function setUp() {  
    parent::setUp();
```

```
    $this->placeBlock('local_tasks_block');
```

```
    $this->placeBlock('local_actions_block');
```

```
    $this->placeBlock('page_title_block');
```

```
    $this->adminUser = $this->drupalCreateUser(['administer advancedqueue']);
```

```
    $this->drupalLogin($this->adminUser);
```

```
}
```

```
// tests/src/Functional/QueueTest.php

public function testQueueDeletion() {
    $queue = Queue::create([
        'id' => 'test',
        'label' => 'Test',
        'backend' => 'database',
        'processor' => QueueInterface::PROCESSOR_DAEMON,
        'processing_time' => 100,
    ]);
    $queue->save();
    $this->drupalGet('admin/config/system/queues/manage/' . $queue->id() . '/delete');
    $this->submitForm([], 'Delete');
    $this->assertSession()->addressEquals('admin/config/system/queues');

    $queue_exists = (bool) Queue::load('test');
    $this->assertEmpty($queue_exists, 'The queue has been deleted from the database.');
```

```
// tests/src/Functional/QueueTest.php

public function testQueueDeletion() {
    $queue = Queue::create([
        'id' => 'test',
        'label' => 'Test',
        'backend' => 'database',
        'processor' => QueueInterface::PROCESSOR_DAEMON,
        'processing_time' => 100,
    ]);
    $queue->save();
    $this->drupalGet('admin/config/system/queues/manage/' . $queue->id() . '/delete');
    $this->submitForm([], 'Delete');
    $this->assertSession()->addressEquals('admin/config/system/queues');

    $queue_exists = (bool) Queue::load('test');
    $this->assertEmpty($queue_exists, 'The queue has been deleted from the database.');
```

```
// tests/src/Functional/QueueTest.php

public function testQueueDeletion() {
    $queue = Queue::create([
        'id' => 'test',
        'label' => 'Test',
        'backend' => 'database',
        'processor' => QueueInterface::PROCESSOR_DAEMON,
        'processing_time' => 100,
    ]);
    $queue->save();
    $this->drupalGet('admin/config/system/queues/manage/' . $queue->id() . '/delete');
    $this->submitForm([], 'Delete');
    $this->assertSession()->addressEquals('admin/config/system/queues');

    $queue_exists = (bool) Queue::load('test');
    $this->assertEmpty($queue_exists, 'The queue has been deleted from the database.');
```

**HOW DO I KNOW**

**WHICH TYPE OF TEST TO USE?**

**NEED A BROWSER?**

**USE A FUNCTIONAL TEST**

**INTERACT WITH OTHER SERVICES?**

**USE A KERNEL TEST**



**ISOLATED PHP CODE?**

**USE A UNIT TEST**

**SHOULD YOU TEST THAT  
A BLOCK IS RENDERED  
CORRECTLY?**

**OR SHOULD YOU TEST  
YOUR RENDER ARRAY TO  
GENERATE THE BLOCK?**

# SETUP (FUNCTIONAL)

drupalCreateUser()

drupalCreateRole()

drupalLogin()

drupalLogout()

drupalGet()

drupalPost()

drupalPostForm()

# SETUP (KERNEL)

```
# UserCreationTrait
```

```
createUser()
```

```
createAdminRole()
```

```
createRole()
```

```
checkPermissions()
```

```
# CommentTestTrait
```

```
addDefaultCommentField()
```

```
# AssertMailTrait
```

```
getMails()
```

```
assertMail()
```

# ASSERTIONS

`assertTrue()`

`assertFalse()`

`assertEquals()`

`assertSame()`

`assertNull()`

`assertNotNull()`

`assertCount()`

`assertEmpty()`

`assertArraySubset()`

# ASSERTIONS (FUNCTIONAL)

`assertSession()`

`pageTextContains()`

`pageTextNotContains()`

`linkByHrefExists()`

`linkByHrefNotExists()`

`statusCodeEquals()`

`statusCodeNotEquals()`

**REAL LIFE**

**EXAMPLE**





broadbean  
technology

# SPECIFICATION

- ▶ **Job adverts created in Broadbean UI, needs to create nodes in Drupal**
- ▶ **Application URL links users to separate application system**
- ▶ **Jobs need to be linked to offices**
- ▶ **Job length specified in number of days**
- ▶ **Path is specified as a field in the API**
- ▶ **Application URL constructed from domain, includes role ID as a GET parameter and optionally UTM parameters**

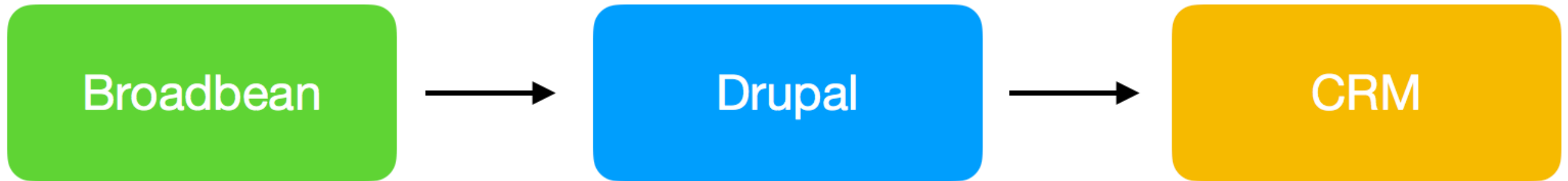
Broadbean



Drupal



CRM





```
$data = [  
  'command' => 'add',  
  'username' => 'bobsmith',  
  'password' => 'p455w0rd',  
  'active_for' => '365',  
  'application_email' => 'bob.12345.123@smith.aplitrak.com',  
  'branch_address' => '123 Fake St, Bristol, BS1 2AB',  
  'branch_name' => 'Test',  
  'contract' => 'Temporary',  
  'details' => 'This is the detailed description.',  
  'job_id' => 'abc123_1234567',  
  'job_title' => 'Healthcare Assistant (HCA)',  
  'job_type' => 'Care at Home',  
  'keywords' => 'flexible, Bristol, part-time',  
  'locations' => 'Bath, Devizes',  
  'role_id' => 'A/52/86',  
  'salary' => '32,000.00 per annum',  
  'salary_prefix' => 'Basic Salary',  
  'status' => 'Part time',  
  'summary' => 'This is the short description.',  
  'url_alias' => 'healthcare-assistant-aldershot-june17',  
];
```

# IMPLEMENTATION

- ▶ Added route to accept data from API as XML
- ▶ Added system user with API role to authenticate
- ▶ **active\_for** converted from number of days to UNIX timestamp
- ▶ **branch\_name** and **locations** converted from plain text to entity reference (job node to office node)
- ▶ **url\_alias** property mapped to **path**

# IMPLEMENTATION

- ▶ **If no error, create the job node, return OK response to Broadbean**
- ▶ **If an Exception is thrown, return an error code and message**

# TESTING GOALS

- ▶ Ensure job nodes are **successfully created**
- ▶ Ensure that fields are **mapped correctly**
- ▶ Ensure that **calculations are correct**
- ▶ Ensure that entity references are **linked correctly**



# TYPES OF TESTS

- ▶ **Unit:** ensure number of days are converted to timestamps correctly
- ▶ **Kernel:** job nodes can be added and deleted, expired job nodes are deleted, application URL is generated correctly
- ▶ **Functional:** job nodes are created with the correct URL and the correct response code is returned
- ▶ **FunctionalJavaScript:** application URL is updated with JavaScript based on UTM parameters (hosting)

# RESULTS

- ▶ **0 bugs!**
- ▶ **Reduced debugging time**
- ▶ **Easier to identify where issues occurred and responsibilities**

# **RUNNING TESTS**

**OPTION 1**

**SIMPLETEST MODULE (UI)**

- ▶ ParamConverter
- ▶ path
- ▶ Path
- ▶ PathProcessor
- ▶ PhpStorage
- ▶ PHPUnit
- ▶ phpunit\_example
- ▶ Plugin
- ▶ plugin\_type\_example
- ▶ PrivateKeyTest
- ▶ proxy\_builder
- ▶ Queue
- ▶ queue\_example
- ▶ quickedit
- ▶ rdf

- utility
- Uuid
- validation
- Validation
- Variable
- views
- Views
- views\_ui
- workflows

Run tests

**CLEAN TEST ENVIRONMENT**  
[Clean environment](#)  
Remove tables with the prefix "test" followed by digits and temporary directories that are left over from tests that crashed. This is intended for developers when creating tests.

# Running tests ☆



## Running tests ☆

Processed test 1 of 4 - *Drupal\Tests\phpunit\_example\Functional\PHPUnitExampleMenuTest.*

Overall results: 2 passes, 0 fails, 0 exceptions

- *Drupal\Tests\phpunit\_example\Functional\PHPUnitExampleMenuTest*: 2 passes, 0 fails, 0 exceptions





## Running tests ☆

Processed test 4 of 4 - *Drupal\Tests\phpunit\_example\Unit\ProtectedPrivatesTest*.

Overall results: 34 passes, 0 fails, 0 exceptions

- *Drupal\Tests\phpunit\_example\Unit\ProtectedPrivatesTest*: 4 passes, 0 fails, 0 exceptions
- *Drupal\Tests\phpunit\_example\Unit\DisplayManagerTest*: 1 pass, 0 fails, 0 exceptions
- *Drupal\Tests\phpunit\_example\Unit\AddClassTest*: 27 passes, 0 fails, 0 exceptions
- *Drupal\Tests\phpunit\_example\Functional\PHPUnitExampleMenuTest*: 2 passes, 0 fails, 0 exceptions



100%

## Test result

[Home](#) » [Administration](#) » [Configuration](#) » [Development](#) » [Testing](#)

✓ The test run finished in 1 min 19 sec.

**ACTIONS**  
Filter: All (4) ▾ Run tests [Return to list](#)

**RESULTS**  
34 passes, 0 fails, 0 exceptions

- ▶ `Drupal\Tests\phpunit_example\Functional\PHPUnitExampleMenuTest`
- ▶ `Drupal\Tests\phpunit_example\Unit\AddClassTest`
- ▶ `Drupal\Tests\phpunit_example\Unit\DisplayManagerTest`
- ▶ `Drupal\Tests\phpunit_example\Unit\ProtectedPrivatesTest`

▼ Drupal\Tests\phpunit\_example\Unit\AddClassTest

AddClass units tests.  
27 passes, 0 fails, 0 exceptions

MESSAGE	GROUP	FILENAME	LINE	FUNCTION	STATUS
	Other	AddClassTest.php	46	Drupal\Tests\phpunit_example\Unit\AddClassTest->testAdd()	✓
	Other	AddClassTest.php	87	Drupal\Tests\phpunit_example\Unit\AddClassTest->testAddWithDataProvider with data set #0()	✓
	Other	AddClassTest.php	87	Drupal\Tests\phpunit_example\Unit\AddClassTest->testAddWithDataProvider with data set #1()	✓
	Other	AddClassTest.php	114	Drupal\Tests\phpunit_example\Unit\AddClassTest->testAddWithBadDataProvider with data set #0()	✓
	Other	AddClassTest.php	114	Drupal\Tests\phpunit_example\Unit\AddClassTest->testAddWithBadDataProvider with data set #1()	✓
	Other	AddClassTest.php	114	Drupal\Tests\phpunit_example\Unit\AddClassTest->testAddWithBadDataProvider with data set #2()	✓
	Other	AddClassTest.php	114	Drupal\Tests\phpunit_example\Unit\AddClassTest->testAddWithBadDataProvider with data set #3()	✓
	Other	AddClassTest.php	114	Drupal\Tests\phpunit_example\Unit\AddClassTest->testAddWithBadDataProvider with data set #4()	✓
	Other	AddClassTest.php	114	Drupal\Tests\phpunit_example\Unit\AddClassTest->testAddWithBadDataProvider with data set #5()	✓
	Other	AddClassTest.php	114	Drupal\Tests\phpunit_example\Unit\AddClassTest->testAddWithBadDataProvider with data set #6()	✓
	Other	AddClassTest.php	114	Drupal\Tests\phpunit_example\Unit\AddClassTest->testAddWithBadDataProvider with data set #7()	✓
	Other	AddClassTest.php	114	Drupal\Tests\phpunit_example\Unit\AddClassTest->testAddWithBadDataProvider with data set #8()	✓

**OPTION 2**

**CORE SCRIPT**

```
$ php core/scripts/run-tests.sh
```

```
$ php core/scripts/run-tests.sh --module example
```

```
$ php core/scripts/run-tests.sh --class ExampleTest
```

**OPTION 3**

**PHPUNIT**

# PREREQUISITE (CREATING A PHPUNIT.XML FILE)

- ▶ **Configures PHPUnit**
- ▶ **Needed to run some types of tests**
- ▶ **Ignored by Git by default**
- ▶ **Copy `core/phpunit.xml.dist` to `core/phpunit.xml`**
- ▶ **Add and change as needed**
  - ▶ `SIMPLETEST_BASE_URL, SIMPLETEST_DB, BROWSERTEST_OUTPUT_DIRECTORY`
  - ▶ `stopOnFailure="true"`

```
cd web
```

```
../vendor/bin/phpunit -c core \  
modules/contrib/examples/phpunit_example
```



```
cd web/core
```

```
../../vendor/bin/phpunit \
```

```
../modules/contrib/examples/phpunit_example
```

--filter

--testsuite

--group

--colors

--stop-on-failure

--verbose --debug

```
drupal-testing-workshop [~/Code/os/drupal-testing-workshop] - .../web/modules/contrib/examples/phpunit_example/tests/src/Functional/PHPUnitExampleMenuTest.php [drupal-testing-workshop]

Project
├── js_example
├── menu_example
├── node_type_example
├── page_example
├── pager_example
├── phpunit_example
│   ├── src
│   └── templates
├── tests
│   └── src
│       ├── Functional
│       │   └── PHPUnitExampleMenuTest.php
│       └── Unit
│           ├── Subclasses
│           │   ├── AddClassTest.php
│           │   ├── DisplayManagerTest.php
│           │   └── ProtectedPrivatesTest.php
│           ├── phpunit_example.info.yml
│           ├── phpunit_example.links.menu.yml
│           ├── phpunit_example.module
│           └── phpunit_example.routing.yml
└── plugin_type_example

57     $this->drupalGet($page);
58     $this->assertLinkByHref($path);
59     }
60 }
61
62 /**
63  * Tests phpunit_example menus.
64  */
65 public function testPhpUnitExampleMenu() {
66     $this->drupalGet('/examples/phpunit-example');
67     $this->assertResponse(200, 'Description page exists.');
```

Run: tests x

Tests passed: 34 of 34 tests – 29 s 240 ms

Time: 46.62 seconds, Memory: 6.00MB

OK (34 tests, 41 assertions)

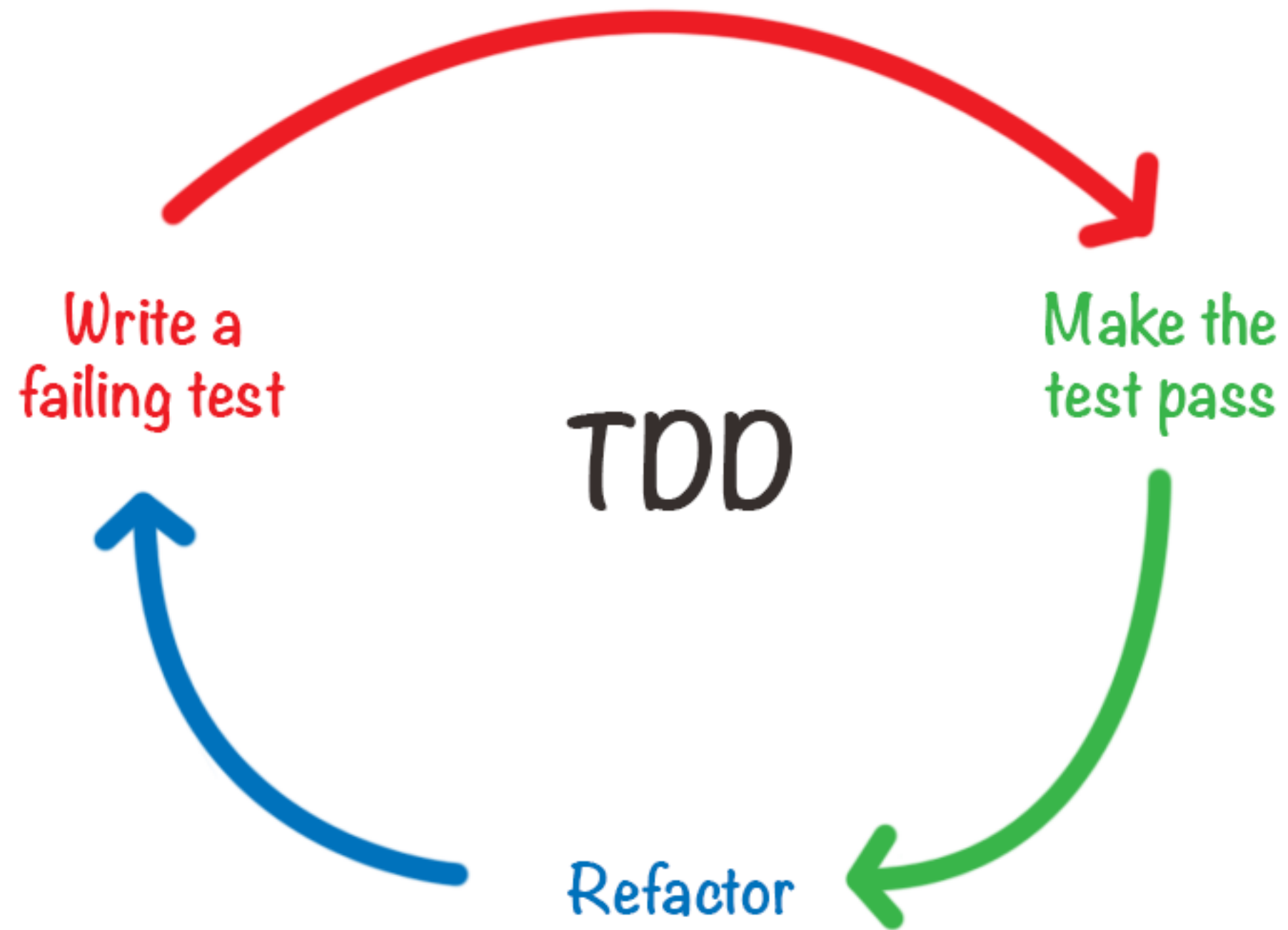
Process finished with exit code 0

- ✓ /var/www/web/modules/contrib/exa 29 s 240 ms
  - ✓ Drupal\Tests\phpunit\_example\Fu 29 s 60 ms
    - ✓ testPhpUnitExampleLink 15 s 570 ms
    - ✓ testPhpUnitExampleMenu 13 s 490 ms
  - ▶ ✓ Drupal\Tests\phpunit\_example\Unit\A 40 ms
  - ✓ Drupal\Tests\phpunit\_example\Unit\I 120 ms
    - ✓ testSimpleMockDisplayManager 120 ms
  - ✓ Drupal\Tests\phpunit\_example\Unit\P 20 ms
    - ▶ ✓ testPrivateAdd 10 ms
    - ▶ ✓ testPrivateAddBadData 0 ms
    - ✓ testProtectedAdd 10 ms
      - ✓ with data set #0 10 ms
    - ✓ testProtectedAddBadData 0 ms
      - ✓ with data set #0 0 ms

**TEST DRIVEN  
DEVELOPMENT**

# TEST DRIVEN DEVELOPMENT

- ▶ **Write a test**
- ▶ **Test fails**
- ▶ **Write code**
- ▶ **Test passes**
- ▶ **Refactor**
- ▶ **Repeat**



**RED, GREEN, REFACTOR**

# PORTING MODULES TO DRUPAL 8

- ▶ **Make a new branch**
- ▶ **Add/update the tests**
- ▶ **Write code to make the tests pass**
- ▶ **Refactor**
- ▶ **Repeat**



# HOW I WRITE TESTS - "OUTSIDE IN"

- ▶ **Start with functional tests**
- ▶ **Drop down to integration or unit tests where needed**
- ▶ **Programming by wishful thinking**
- ▶ **Write comments first, then fill in the code**
- ▶ **Sometimes write assertions first**

# **BUILDING A NEW DRUPAL 8 MODULE WITH TEST DRIVEN DEVELOPMENT**

# ACCEPTANCE CRITERIA

- ▶ **As a site visitor**
- ▶ **I want to see a list of published articles at /blog**
- ▶ **Ordered by post date**

# TASKS

- ▶ **Ensure the blog page exists**
- ▶ **Ensure only published articles are shown**
- ▶ **Ensure the articles are shown in the correct order**

# IMPLEMENTATION

- ▶ **Use views module**
- ▶ **Do the minimum amount at each step, make no assumptions, let the tests guide us**
- ▶ **Start with functional test**

**STEP 1**

**CREATE THE MODULE**

```
# tdd_blog.info.yml
```

```
name: 'TDD Blog'
```

```
core: '8.x'
```

```
type: 'module'
```

**STEP 2**

**ENSURE THE BLOG PAGE  
EXISTS**



```
// tests/src/Functional/BlogPageTest.php

namespace Drupal\Tests\tdd_blog\Functional;

use Drupal\Tests\BrowserTestBase;

class BlogPageTest extends BrowserTestBase {

    protected static $modules = ['tdd_blog'];

    public function testBlogPageExists() {
        $this->drupalGet('/blog');

        $this->assertSession()->statusCodeEquals(200);
    }
}
```

```
// tests/src/Functional/BlogPageTest.php

namespace Drupal\Tests\tdd_blog\Functional;

use Drupal\Tests\BrowserTestBase;

class BlogPageTest extends BrowserTestBase {

    protected static $modules = ['tdd_blog'];

    public function testBlogPageExists() {
        $this->drupalGet('/blog');

        $this->assertSession()->statusCodeEquals(200);
    }
}
```

```
// tests/src/Functional/BlogPageTest.php

namespace Drupal\Tests\tdd_blog\Functional;

use Drupal\Tests\BrowserTestBase;

class BlogPageTest extends BrowserTestBase {

    protected static $modules = ['tdd_blog'];

    public function testBlogPageExists() {
        $this->drupalGet('/blog');

        $this->assertSession()->statusCodeEquals(200);
    }

}
```

```
// tests/src/Functional/BlogPageTest.php

namespace Drupal\Tests\tdd_blog\Functional;

use Drupal\Tests\BrowserTestBase;

class BlogPageTest extends BrowserTestBase {

    protected static $modules = ['tdd_blog'];

    public function testBlogPageExists() {
        $this->drupalGet('/blog');

        $this->assertSession()->statusCodeEquals(200);
    }
}
```

```
// tests/src/Functional/BlogPageTest.php

namespace Drupal\Tests\tdd_blog\Functional;

use Drupal\Tests\BrowserTestBase;

class BlogPageTest extends BrowserTestBase {

    protected static $modules = ['tdd_blog'];

    public function testBlogPageExists() {
        $this->drupalGet('/blog');

        $this->assertSession()->statusCodeEquals(200);
    }
}
```

```
docker@cli:/var/www/web$ ../vendor/bin/phpunit -c core/modules/custom/tdd_blog
PHPUnit 6.5.8 by Sebastian Bergmann and contributors.
```

```
Testing modules/custom/tdd_blog
```

```
E                                                                    1 / 1 (100%)
```

```
Time: 19.31 seconds, Memory: 6.00MB
```

```
There was 1 error:
```

```
1) Drupal\Tests\tdd_blog\Functional\BlogPageTest::testBlogPageExists
```

```
Behat\Mink\Exception\ExpectationException: Current response status code is 404, but 200 expected.
```

```
/var/www/vendor/behat/mink/src/WebAssert.php:768
```

```
/var/www/vendor/behat/mink/src/WebAssert.php:130
```

```
/var/www/web/modules/custom/tdd_blog/tests/src/Functional/BlogPageTest.php:13
```

```
ERRORS!
```

```
Tests: 1, Assertions: 3, Errors: 1.
```

```
docker@cli:/var/www/web$ ../vendor/bin/phpunit -c core modules/custom/tdd_blog
PHPUnit 6.5.8 by Sebastian Bergmann and contributors.
```

```
Testing modules/custom/tdd_blog
```

```
E                                                                    1 / 1 (100%)
```

```
Time: 19.31 seconds, Memory: 6.00MB
```

```
There was 1 error:
```

```
1) Drupal\Tests\tdd_blog\Functional\BlogPageTest::testBlogPageExists
```

```
Behat\Mink\Exception\ExpectationException: Current response status code is 404, but 200 expected.
```

```
/var/www/vendor/behat/mink/src/WebAssert.php:768
```

```
/var/www/vendor/behat/mink/src/WebAssert.php:130
```

```
/var/www/web/modules/custom/tdd_blog/tests/src/Functional/BlogPageTest.php:13
```

```
ERRORS!
```

```
Tests: 1, Assertions: 3, Errors: 1.
```

```
docker@cli:/var/www/web$ ../vendor/bin/phpunit -c core modules/custom/tdd_blog
PHPUnit 6.5.8 by Sebastian Bergmann and contributors.
```

```
Testing modules/custom/tdd_blog
```

```
E 1 / 1 (100%)
```

```
Time: 19.31 seconds, Memory: 6.00MB
```

```
There was 1 error:
```

```
1) Drupal\Tests\tdd_blog\Functional\BlogPageTest::testBlogPageExists
```

```
Behat\Mink\Exception\ExpectationException: Current response status code is 404, but 200 expected.
```

```
/var/www/vendor/behat/mink/src/WebAssert.php:768
```

```
/var/www/vendor/behat/mink/src/WebAssert.php:130
```

```
/var/www/web/modules/custom/tdd_blog/tests/src/Functional/BlogPageTest.php:13
```

```
ERRORS!
```

```
Tests: 1, Assertions: 3, Errors: 1.
```



```
docker@cli:/var/www/web$ ../vendor/bin/phpunit -c core modules/custom/tdd_blog
PHPUnit 6.5.8 by Sebastian Bergmann and contributors.
```

```
Testing modules/custom/tdd_blog
```

```
E                                                                    1 / 1 (100%)
```

```
Time: 19.31 seconds, Memory: 6.00MB
```

```
There was 1 error:
```

```
1) Drupal\Tests\tdd_blog\Functional\BlogPageTest::testBlogPageExists
```

```
Behat\Mink\Exception\ExpectationException: Current response status code is 404, but 200 expected.
```

```
/var/www/vendor/behat/mink/src/WebAssert.php:768
```

```
/var/www/vendor/behat/mink/src/WebAssert.php:130
```

```
/var/www/web/modules/custom/tdd_blog/tests/src/Functional/BlogPageTest.php:13
```

```
ERRORS!
```

```
Tests: 1, Assertions: 3, Errors: 1.
```

```
docker@cli:/var/www/web$ ../vendor/bin/phpunit -c core modules/custom/tdd_blog
PHPUnit 6.5.8 by Sebastian Bergmann and contributors.
```

```
Testing modules/custom/tdd_blog
```

```
E                                                                    1 / 1 (100%)
```

```
Time: 19.31 seconds, Memory: 6.00MB
```

```
There was 1 error:
```

```
1) Drupal\Tests\tdd_blog\Functional\BlogPageTest::testBlogPageExists
```

```
Behat\Mink\Exception\ExpectationException: Current response status code is 404, but 200 expected.
```

```
/var/www/vendor/behat/mink/src/WebAssert.php:768
```

```
/var/www/vendor/behat/mink/src/WebAssert.php:130
```

```
/var/www/web/modules/custom/tdd_blog/tests/src/Functional/BlogPageTest.php:13
```

```
ERRORS!
```

```
Tests: 1, Assertions: 3, Errors: 1.
```

- ▶ **The view has not been created**
- ▶ **Create a new view, page display**
- ▶ **Set the path**
- ▶ **Export the config**
- ▶ **Copy it into the module's `config/install` directory**

## Add view

[Home](#) » [Administration](#) » [Structure](#) » [Views](#)

### VIEW BASIC INFORMATION

View name \*

Machine name: `blog` [\[Edit\]](#)

Description

### VIEW SETTINGS

Show:  of type:  tagged with:  sorted by:

### PAGE SETTINGS

Create a page

### BLOCK SETTINGS

Create a block

# Blog (Content) ☆

Home » Administration » Structure » Views » Blog » Edit

## Displays

Page + Add Edit view name/description

Display name: Page View Page

<b>TITLE</b> Title: None	<b>PAGE SETTINGS</b> Path: /blog Menu: No menu Access: Permission   View published content	<b>ADVANCED</b>
<b>FORMAT</b> Format: Unformatted list   Settings Show: Fields   Settings	<b>HEADER</b> Add	
<b>FIELDS</b> Add	<b>FOOTER</b> Add	
<b>FILTER CRITERIA</b> Add	<b>NO RESULTS BEHAVIOR</b> Add	
<b>SORT CRITERIA</b> Add	<b>PAGER</b> Use pager: Mini   Mini pager, 10 items More link: No	

Save Cancel

```
drush cex -y
```

```
cp ../config/default/views.view.blog.yml \  
modules/custom/tdd_blog/config/install
```

```
# views.view.blog.yml

- uuid: 84305edf-7aef-4109-bc93-e87f685fb678
langcode: en
status: true
dependencies:
  config:
    - node.type.article
  module:
    - node
    - user
- _core:
- default_config_hash: iGZkqLWpwWN0Rq6_fy6v_Kn_KE4BjYHqj9vpgQsWJCs
id: blog
...
```

```
docker@cli:/var/www/web$ ../vendor/bin/phpunit -c core/modules/custom/tdd_blog
PHPUnit 6.5.8 by Sebastian Bergmann and contributors.
```

```
Testing modules/custom/tdd_blog
```

```
E 1 / 1 (100%)
```

```
Time: 16.02 seconds, Memory: 6.00MB
```

```
There was 1 error:
```

```
1) Drupal\Tests\tdd_blog\Functional\BlogPageTest::testBlogPageExists
```

```
Drupal\Core\Config\UnmetDependenciesException: Configuration objects provided by <em class="placeholder">tdd_blog</em>
have unmet dependencies: <em class="placeholder">views.view.blog (node.type.article, node, views)</em>
```

```
/var/www/web/core/lib/Drupal/Core/Config/UnmetDependenciesException.php:98
```

```
/var/www/web/core/lib/Drupal/Core/Config/ConfigInstaller.php:469
```

```
/var/www/web/core/lib/Drupal/Core/ProxyClass/Config/ConfigInstaller.php:132
```

```
/var/www/web/core/lib/Drupal/Core/Extension/ModuleInstaller.php:145
```

```
/var/www/web/core/lib/Drupal/Core/ProxyClass/Extension/ModuleInstaller.php:83
```

```
/var/www/web/core/lib/Drupal/Core/Test/FunctionalTestSetupTrait.php:437
```

```
/var/www/web/core/tests/Drupal/Tests/BrowserTestBase.php:1055
```

```
/var/www/web/core/tests/Drupal/Tests/BrowserTestBase.php:490
```

```
ERRORS!
```

```
Tests: 1, Assertions: 0, Errors: 1.
```



```
# tdd_blog.info.yml
```

```
name: 'TDD Blog'
```

```
description: 'A demo module to show test driven module development.'
```

```
core: 8.x
```

```
type: module
```

```
dependencies:
```

- 'drupal:node'
- 'drupal:views'

```
docker@cli:/var/www/web$ ../vendor/bin/phpunit -c core/modules/custom/tdd_blog
PHPUnit 6.5.8 by Sebastian Bergmann and contributors.
```

```
Testing modules/custom/tdd_blog
```

```
E 1 / 1 (100%)
```

```
Time: 20 seconds, Memory: 6.00MB
```

```
There was 1 error:
```

```
1) Drupal\Tests\tdd_blog\Functional\BlogPageTest::testBlogPageExists
```

```
Drupal\Core\Config\UnmetDependenciesException: Configuration objects provided by <em class="placeholder">tdd_blog</em>
have unmet dependencies: <em class="placeholder">views.view.blog (node.type.article)</em>
```

```
/var/www/web/core/lib/Drupal/Core/Config/UnmetDependenciesException.php:98
```

```
/var/www/web/core/lib/Drupal/Core/Config/ConfigInstaller.php:469
```

```
/var/www/web/core/lib/Drupal/Core/ProxyClass/Config/ConfigInstaller.php:132
```

```
/var/www/web/core/lib/Drupal/Core/Extension/ModuleInstaller.php:145
```

```
/var/www/web/core/lib/Drupal/Core/ProxyClass/Extension/ModuleInstaller.php:83
```

```
/var/www/web/core/lib/Drupal/Core/Test/FunctionalTestSetupTrait.php:437
```

```
/var/www/web/core/tests/Drupal/Tests/BrowserTestBase.php:1055
```

```
/var/www/web/core/tests/Drupal/Tests/BrowserTestBase.php:490
```

```
ERRORS!
```

```
Tests: 1, Assertions: 0, Errors: 1.
```

▶ **Add the article content type**

```
docker@cli:/var/www/web$ ../vendor/bin/phpunit -c core modules/custom/tdd_blog
PHPUnit 6.5.8 by Sebastian Bergmann and contributors.
```

```
Testing modules/custom/tdd_blog
```

```
. 1 / 1 (100%)
```

```
Time: 23.36 seconds, Memory: 6.00MB
```

```
OK (1 test, 3 assertions)
```

# TASKS

- ▶ ~~Ensure the blog page exists~~
- ▶ Ensure only published articles are shown
- ▶ Ensure the articles are shown in the correct order

## STEP 3

**ENSURE ONLY PUBLISHED  
ARTICLES ARE SHOWN**

```
public function testOnlyPublishedArticlesAreShown() {  
    // Given I have a mixture of published and unpublished articles,  
    // as well as other types of content.  
  
    // When I view the blog page.  
  
    // I should only see the published articles.  
}
```

**OPTION 1**

**FUNCTIONAL TESTS**



```
// modules/custom/tdd_blog/tests/src/Functional/BlogPageTest.php

public function testOnlyPublishedArticlesAreShown() {
    // Given I have a mixture of published and unpublished articles,
    // as well as other types of content.
    $node1 = $this->drupalCreateNode(['type' => 'page', 'status' => 1]);
    $node2 = $this->drupalCreateNode(['type' => 'article', 'status' => 1]);
    $node3 = $this->drupalCreateNode(['type' => 'article', 'status' => 0]);

    // When I view the blog page.
    $this->drupalGet('/blog');

    // I should only see the published articles.
    $assert = $this->assertSession();
    $assert->pageTextContains($node2->label());
    $assert->pageTextNotContains($node1->label());
    $assert->pageTextNotContains($node3->label());
}
```

```
// modules/custom/tdd_blog/tests/src/Functional/BlogPageTest.php

public function testOnlyPublishedArticlesAreShown() {
    // Given I have a mixture of published and unpublished articles,
    // as well as other types of content.
    $node1 = $this->drupalCreateNode(['type' => 'page', 'status' => 1]);
    $node2 = $this->drupalCreateNode(['type' => 'article', 'status' => 1]);
    $node3 = $this->drupalCreateNode(['type' => 'article', 'status' => 0]);

    // When I view the blog page.
    $this->drupalGet('/blog');

    // I should only see the published articles.
    $assert = $this->assertSession();
    $assert->pageTextContains($node2->label());
    $assert->pageTextNotContains($node1->label());
    $assert->pageTextNotContains($node3->label());
}
```

```
// modules/custom/tdd_blog/tests/src/Functional/BlogPageTest.php
```

```
public function testOnlyPublishedArticlesAreShown() {  
    // Given I have a mixture of published and unpublished articles,  
    // as well as other types of content.  
    $node1 = $this->drupalCreateNode(['type' => 'page', 'status' => 1]);  
    $node2 = $this->drupalCreateNode(['type' => 'article', 'status' => 1]);  
    $node3 = $this->drupalCreateNode(['type' => 'article', 'status' => 0]);  
  
    // When I view the blog page.  
    $this->drupalGet('/blog');  
  
    // I should only see the published articles.  
    $assert = $this->assertSession();  
    $assert->pageTextContains($node2->label());  
    $assert->pageTextNotContains($node1->label());  
    $assert->pageTextNotContains($node3->label());  
}
```

**OPTION 2**

**KERNEL TESTS**

```
namespace Drupal\Tests\tdd_blog\Kernel;

use Drupal\KernelTests\Core\Entity\EntityKernelTestBase;
use Drupal\Tests\node\Traits\NodeCreationTrait;

class BlogPageTest extends EntityKernelTestBase {

    use NodeCreationTrait;

    public static $modules = ['node'];

    public function testOnlyPublishedArticlesAreShown() {
        $this->createNode(['type' => 'page', 'status' => 1]);
        $this->createNode(['type' => 'article', 'status' => 1]);
        $this->createNode(['type' => 'article', 'status' => 0]);
    }

}
```

```
namespace Drupal\Tests\tdd_blog\Kernel;

use Drupal\KernelTests\Core\Entity\EntityKernelTestBase;
use Drupal\Tests\node\Traits\NodeCreationTrait;

class BlogPageTest extends EntityKernelTestBase {

    use NodeCreationTrait;

    public static $modules = ['node'];

    public function testOnlyPublishedArticlesAreShown() {
        $this->createNode(['type' => 'page', 'status' => 1]);
        $this->createNode(['type' => 'article', 'status' => 1]);
        $this->createNode(['type' => 'article', 'status' => 0]);
    }

}
```



```
namespace Drupal\Tests\tdd_blog\Kernel;

use Drupal\KernelTests\Core\Entity\EntityKernelTestBase;
use Drupal\Tests\node\Traits\NodeCreationTrait;

class BlogPageTest extends EntityKernelTestBase {

    use NodeCreationTrait;

    public static $modules = ['node'];

    public function testOnlyPublishedArticlesAreShown() {
        $this->createNode(['type' => 'page', 'status' => 1]);
        $this->createNode(['type' => 'article', 'status' => 1]);
        $this->createNode(['type' => 'article', 'status' => 0]);
    }

}
```



```
docker@cli:/var/www/web$ ../vendor/bin/phpunit -c core/modules/custom/tdd_blog/tests/src/Kernel/
PHPUnit 6.5.8 by Sebastian Bergmann and contributors.
```

```
Testing modules/custom/tdd_blog/tests/src/Kernel/
```

```
E                                                                    1 / 1 (100%)
```

```
Time: 6.22 seconds, Memory: 6.00MB
```

There was 1 error:

1) Drupal\Tests\tdd\_blog\Kernel\BlogPageTest::testOnlyPublishedArticlesAreShown

Error: Call to a member function id() on boolean

/var/www/web/core/modules/filter/filter.module:212

/var/www/web/core/modules/node/tests/src/Traits/NodeCreationTrait.php:73

/var/www/web/modules/custom/tdd\_blog/tests/src/Kernel/BlogPageTest.php:13

ERRORS!

Tests: 1, Assertions: 2, Errors: 1.

```
public function testOnlyPublishedArticlesAreShown() {  
    $this->installConfig(['filter']);  
  
    $this->createNode(['type' => 'page', 'status' => 1]);  
    $this->createNode(['type' => 'article', 'status' => 1]);  
    $this->createNode(['type' => 'article', 'status' => 0]);  
}
```

```
public function testOnlyPublishedArticlesAreShown() {  
    $this->installConfig(['filter']);  
  
    $this->createNode(['type' => 'page', 'status' => 1]);  
    $this->createNode(['type' => 'article', 'status' => 1]);  
    $this->createNode(['type' => 'article', 'status' => 0]);  
  
    $results = views_get_view_result('blog');  
}
```

...

```
public static $modules = ['node', 'tdd_blog', 'views'];

public function testOnlyPublishedArticlesAreShown() {
    $this->installConfig(['filter', 'tdd_blog']);

    $this->createNode(['type' => 'page', 'status' => 1]);
    $this->createNode(['type' => 'article', 'status' => 1]);
    $this->createNode(['type' => 'article', 'status' => 0]);

    $results = views_get_view_result('blog');

    $this->assertCount(1, $results);
    $this->assertEquals(2, $results[0]->_entity->id());
}
```

```
...

public static $modules = ['node', 'tdd_blog', 'views'];

public function testOnlyPublishedArticlesAreShown() {
    $this->installConfig(['filter', 'tdd_blog']);

    $this->createNode(['type' => 'page', 'status' => 1]);
    $this->createNode(['type' => 'article', 'status' => 1]);
    $this->createNode(['type' => 'article', 'status' => 0]);

    $results = views_get_view_result('blog');

    $this->assertCount(1, $results);
    $this->assertEquals(2, $results[0]->_entity->id());
}
```

```
...

public static $modules = ['node', 'tdd_blog', 'views'];

public function testOnlyPublishedArticlesAreShown() {
    $this->installConfig(['filter', 'tdd_blog']);

    $this->createNode(['type' => 'page', 'status' => 1]);
    $this->createNode(['type' => 'article', 'status' => 1]);
    $this->createNode(['type' => 'article', 'status' => 0]);

    $results = views_get_view_result('blog');

    $this->assertCount(1, $results);
    $this->assertEquals(2, $results[0]->_entity->id());
}
```

PHPUnit 6.5.8 by Sebastian Bergmann and contributors.

Testing web/modules/custom/tdd\_blog/tests/src/Kernel

F 1 / 1 (100%)

Time: 2.16 seconds, Memory: 6.00MB

There was 1 failure:

1) Drupal\Tests\tdd\_blog\Kernel\BlogPageTest::testOnlyPublishedArticlesAreShown  
Failed asserting that actual size 2 matches expected size 1.

/Users/opdavies/Code/drupal-testing-workshop/web/modules/custom/tdd\_blog/tests/src/Kernel/BlogPageTest.php:23

FAILURES!

Tests: 1, Assertions: 4, Failures: 1.

PHPUnit 6.5.8 by Sebastian Bergmann and contributors.

Testing web/modules/custom/tdd\_blog/tests/src/Kernel

F 1 / 1 (100%)

Time: 2.16 seconds, Memory: 6.00MB

There was 1 failure:

1) Drupal\Tests\tdd\_blog\Kernel\BlogPageTest::testOnlyPublishedArticlesAreShown  
Failed asserting that actual size 2 matches expected size 1.

/Users/opdavies/Code/drupal-testing-workshop/web/modules/custom/tdd\_blog/tests/src/Kernel/BlogPageTest.php:23

FAILURES!

Tests: 1, Assertions: 4, Failures: 1.



## TITLE


Title: [None](#)

## FORMAT

Format: [Unformatted list](#) | [Settings](#)


Show: [Fields](#) | [Settings](#)

## FIELDS

Add 

Content: [Title](#)

## FILTER CRITERIA

Add 

Content: [Published \(= Yes\)](#)

## SORT CRITERIA

Add

- ▶ **There is no content type filter on the view**
- ▶ **Add the filter**
- ▶ **Re-export and save the view**

## TITLE

Title: [None](#)

## FORMAT

Format: [Unformatted list](#) | [Settings](#)

Show: [Fields](#) | [Settings](#)

## FIELDS

Add



[Content: Title](#)

## FILTER CRITERIA

Add



[Content: Published \(= Yes\)](#)

[Content: Content type \(= Article\)](#)

## SORT CRITERIA

Add

PHPUnit 6.5.8 by Sebastian Bergmann and contributors.

Testing web/modules/custom/tdd\_blog/tests/src/Kernel

. 1 / 1 (100%)

Time: 2.02 seconds, Memory: 6.00MB

OK (1 test, 6 assertions)

# TASKS

- ▶ ~~Ensure the blog page exists~~
- ▶ ~~Ensure only published articles are shown~~
- ▶ Ensure the articles are shown in the correct order

## **STEP 4**

**ENSURE THE ARTICLES ARE  
ORDERED BY DATE**

```
// modules/custom/tdd_blog/tests/src/Kernel/BlogPageTest.php
```

```
public function testArticlesAreOrderedByDate() {
```

```
    // Given that I have numerous articles with different post dates.
```

```
    // When I go to the blog page.
```

```
    // The articles are ordered by post date.
```

```
}
```

```
// modules/custom/tdd_blog/tests/src/Kernel/BlogPageTest.php
```

```
public function testArticlesAreOrderedByDate() {
```

```
    // Given that I have numerous articles with different post dates.
```

```
    $this->createNode(['type' => 'article', 'created' => (new \DateTime())->modify('+1 day')->getTimestamp()]);
```

```
    $this->createNode(['type' => 'article', 'created' => (new \DateTime())->modify('+1 month')->getTimestamp()]);
```

```
    $this->createNode(['type' => 'article', 'created' => (new \DateTime())->modify('+3 days')->getTimestamp()]);
```

```
    $this->createNode(['type' => 'article', 'created' => (new \DateTime())->modify('+1 hour')->getTimestamp()]);
```

```
    // When I go to the blog page.
```

```
    // The articles are ordered by post date.
```

```
}
```



```
$this->createNode([  
    'type' => 'article',  
    'created' => (new \DateTime())->modify('+1 day')->getTimestamp(),  
]);
```

```
// modules/custom/tdd_blog/tests/src/Kernel/BlogPageTest.php

public function testArticlesAreOrderedByDate() {
    // Given that I have numerous articles with different post dates.
    $this->createNode(['type' => 'article', 'created' => (new \DateTime())->modify('+1 day')->getTimestamp()]);
    $this->createNode(['type' => 'article', 'created' => (new \DateTime())->modify('+1 month')->getTimestamp()]);
    $this->createNode(['type' => 'article', 'created' => (new \DateTime())->modify('+3 days')->getTimestamp()]);
    $this->createNode(['type' => 'article', 'created' => (new \DateTime())->modify('+1 hour')->getTimestamp()]);

    // When I go to the blog page.
    $results = views_get_view_result('blog');

    // The articles are ordered by post date.
}
```

```
// modules/custom/tdd_blog/tests/src/Kernel/BlogPageTest.php

public function testArticlesAreOrderedByDate() {
    // Given that I have numerous articles with different post dates.
    $this->createNode(['type' => 'article', 'created' => (new \DateTime())->modify('+1 day')->getTimestamp()]);
    $this->createNode(['type' => 'article', 'created' => (new \DateTime())->modify('+1 month')->getTimestamp()]);
    $this->createNode(['type' => 'article', 'created' => (new \DateTime())->modify('+3 days')->getTimestamp()]);
    $this->createNode(['type' => 'article', 'created' => (new \DateTime())->modify('+1 hour')->getTimestamp()]);

    // When I go to the blog page.
    $results = views_get_view_result('blog');

    $nids = array_map(function(ResultRow $result) {
        return $result->_entity->id();
    }, $results);

    // The articles are ordered by post date.
}
```

```
// modules/custom/tdd_blog/tests/src/Kernel/BlogPageTest.php
```

```
public function testArticlesAreOrderedByDate() {
```

```
    // Given that I have numerous articles with different post dates.
```

```
    $this->createNode(['type' => 'article', 'created' => (new \DateTime())->modify('+1 day')->getTimestamp()]);
```

```
    $this->createNode(['type' => 'article', 'created' => (new \DateTime())->modify('+1 month')->getTimestamp()]);
```

```
    $this->createNode(['type' => 'article', 'created' => (new \DateTime())->modify('+3 days')->getTimestamp()]);
```

```
    $this->createNode(['type' => 'article', 'created' => (new \DateTime())->modify('+1 hour')->getTimestamp()]);
```

```
    // When I go to the blog page.
```

```
    $results = views_get_view_result('blog');
```

```
    $nids = array_map(function(ResultRow $result) {
```

```
        return $result->_entity->id();
```

```
    }, $results);
```

```
    // The articles are ordered by post date.
```

```
    $this->assertEquals([4, 1, 3, 2], $nids);
```

```
}
```

PHPUnit 6.5.8 by Sebastian Bergmann and contributors.

Testing web/modules/custom/tdd\_blog/tests/src/Kernel

F 1 / 1 (100%)

Time: 1.42 seconds, Memory: 6.00MB

There was 1 failure:

1) Drupal\Tests\tdd\_blog\Kernel\BlogPageTest::testArticlesAreOrderedByDate

Failed asserting that two arrays are equal.

--- Expected

+++ Actual

@@ @@

Array (

- 0 => 4

- 1 => 1

- 2 => 3

- 3 => 2

+ 0 => '1'

+ 1 => '2'

+ 2 => '3'

+ 3 => '4'

/Users/opdavies/Code/drupal-testing-workshop/web/core/tests/Drupal/KernelTests/KernelTestBase.php:1114

/Users/opdavies/Code/drupal-testing-workshop/web/modules/custom/tdd\_blog/tests/src/Kernel/BlogPageTest.php:43

FAILURES!

Tests: 1, Assertions: 4, Failures: 1.

PHPUnit 6.5.8 by Sebastian Bergmann and contributors.

Testing web/modules/custom/tdd\_blog/tests/src/Kernel

F

1 / 1 (100%)

Time: 1.42 seconds, Memory: 6.00MB

There was 1 failure:

1) Drupal\Tests\tdd\_blog\Kernel\BlogPageTest::testArticlesAreOrderedByDate

Failed asserting that two arrays are equal.

--- Expected

+++ Actual

@@ @@

Array (

- 0 => 4

- 1 => 1

- 2 => 3

- 3 => 2

+ 0 => '1'

+ 1 => '2'

+ 2 => '3'

+ 3 => '4'

/Users/opdavies/Code/drupal-testing-workshop/web/core/tests/Drupal/KernelTests/KernelTestBase.php:1114

/Users/opdavies/Code/drupal-testing-workshop/web/modules/custom/tdd\_blog/tests/src/Kernel/BlogPageTest.php:43

FAILURES!

Tests: 1, Assertions: 4, Failures: 1.

## TITLE

Title: [None](#)

## FORMAT

Format: [Unformatted list](#) | [Settings](#)

Show: [Fields](#) | [Settings](#)

## FIELDS

Add



[Content: Title](#)

## FILTER CRITERIA

Add



[Content: Published \(= Yes\)](#)

[Content: Content type \(= Article\)](#)

## SORT CRITERIA

Add

- ▶ **There is no sort order defined on the view**
- ▶ **Add the sort order**
- ▶ **Re-export the view**



## TITLE

Title: [None](#)

## FORMAT

Format: [Unformatted list](#) | [Settings](#)

Show: [Fields](#) | [Settings](#)

## FIELDS

Add



[Content: Title](#)

## FILTER CRITERIA

Add



[Content: Published \(= Yes\)](#)

[Content: Content type \(= Article\)](#)

## SORT CRITERIA

Add



[Content: Authored on \(desc\)](#)

PHPUnit 6.5.8 by Sebastian Bergmann and contributors.

Testing web/modules/custom/tdd\_blog/tests/src/Kernel

. 1 / 1 (100%)

Time: 1.74 seconds, Memory: 6.00MB

OK (1 test, 5 assertions)

# TASKS

- ▶ ~~Ensure the blog page exists~~
- ▶ ~~Ensure only published articles are shown~~
- ▶ ~~Ensure the articles are shown in the correct order~~

- folder tdd\_blog
  - folder config
    - folder install
      - node.type.article.yml
      - views.view.blog.yml
    - tdd\_blog.info.yml
  - folder tests
    - folder src
      - folder Functional
        - BlogPageTest.php
      - folder Kernel
        - BlogPageTest.php



[Drupal Testing Workshop](#)

[Blog post 1](#)

[Blog post 2](#)

[Blog post 4](#)

## Tools

- [Add content](#)

## Administration

- [Administration](#)



**TAKE AWAYS**

- ▶ Testing has made me a **better developer**
- ▶ Testing can produce **better quality code**
- ▶ Use the **right type of test** for the right situation
- ▶ Use the **right base class**, use available **traits**
- ▶ Writing tests is an **investment**
- ▶ OK to **start small**, introduce tests gradually
- ▶ Easier to **refactor**
- ▶ Tests can pass, but things can **still be broken**. Tests only report on what they cover.



**HAVING TESTS DOES NOT MEAN  
THERE WILL BE NO BUGS**

**YOU MIGHT BE TESTING THE WRONG THING**  
**MAYBE IT DOESN'T WORK THE**  
**WAY YOU THINK IT DOES**

**HAVE YOU WRITTEN ENOUGH ASSERTIONS?**

**HAVE YOU ONLY COVERED THE  
'HAPPY PATH' SCENARIOS?**

**OTHER MODULES CAN AFFECT THINGS**

**TESTS MAY PASS, BUT FAIL  
WHEN OTHER MODULES ARE  
ENABLED**

**TESTING MAY ADD TIME NOW  
BUT SAVE MORE TIME IN THE FUTURE**

**HOW DO YOU GET QUICKER AT WRITING TESTS?  
BY WRITING MORE TESTS**

**START SMALL**

**SOME TESTS ARE BETTER**

**THAN NO TESTS**



**Oliver Davies**

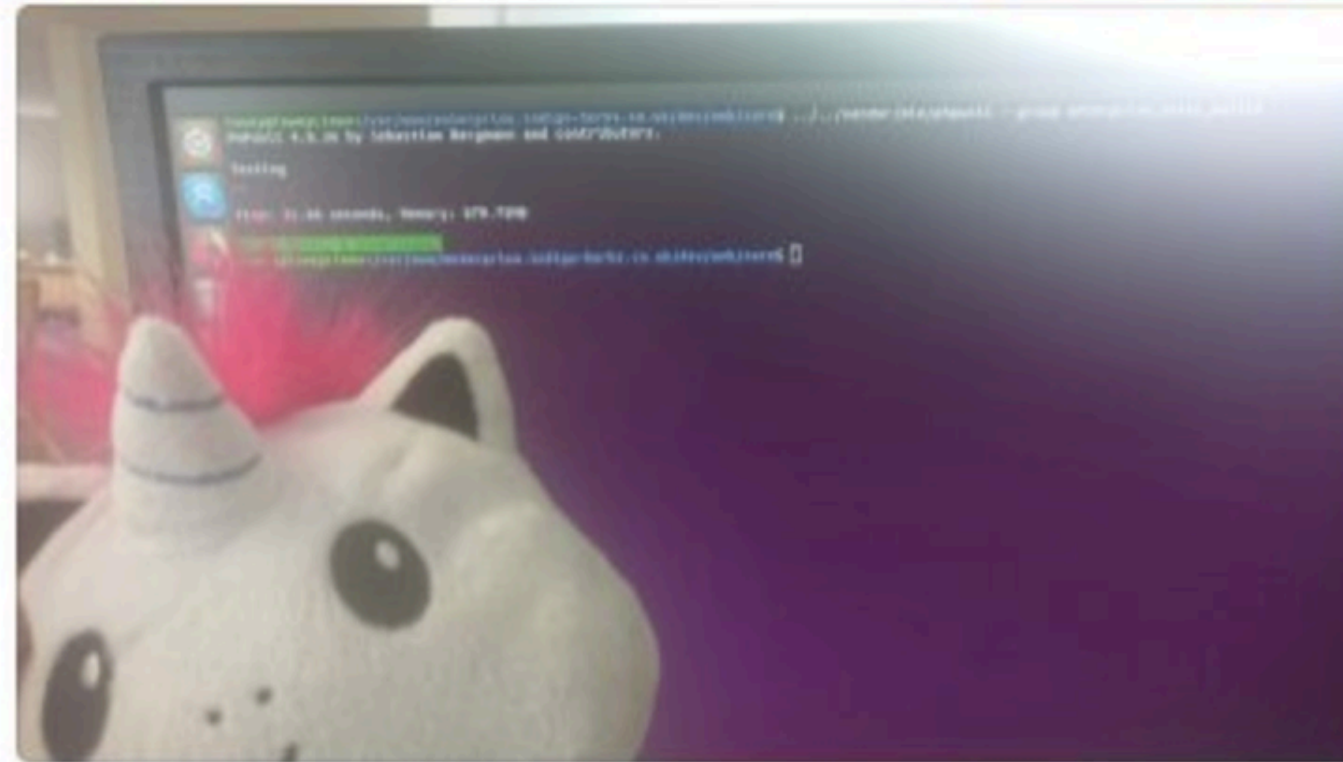
@opdavies



Made my day. 😊 @littlepixiez  
@drupalsomerset

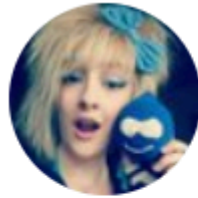
**Tawny Bartlett** Woo tests!!

**Tawny Bartlett** uploaded this image: [IMG\\_20180619\\_153539.jpg](#) ▼



7:41 PM - 19 Jun 2018





**Tawny Bartlett**

@littlepixiez

Following



Replying to [@opdavies](#) [@drupalsomerset](#)

Terrible quality photo but the talk drove me to test and it's great so far, it's really helped give me peace of mind and helped me uncover a bug I wouldn't have otherwise :) so thank you 😊

10:41 PM - 19 Jun 2018

**WHEN YOU DO THINGS RIGHT**

**PEOPLE WON'T BE SURE YOU'VE  
DONE ANYTHING AT ALL**

**QUESTIONS?**

**THANKS**

**@OPDAVIES**

**OLIVERDAVIES.UK**