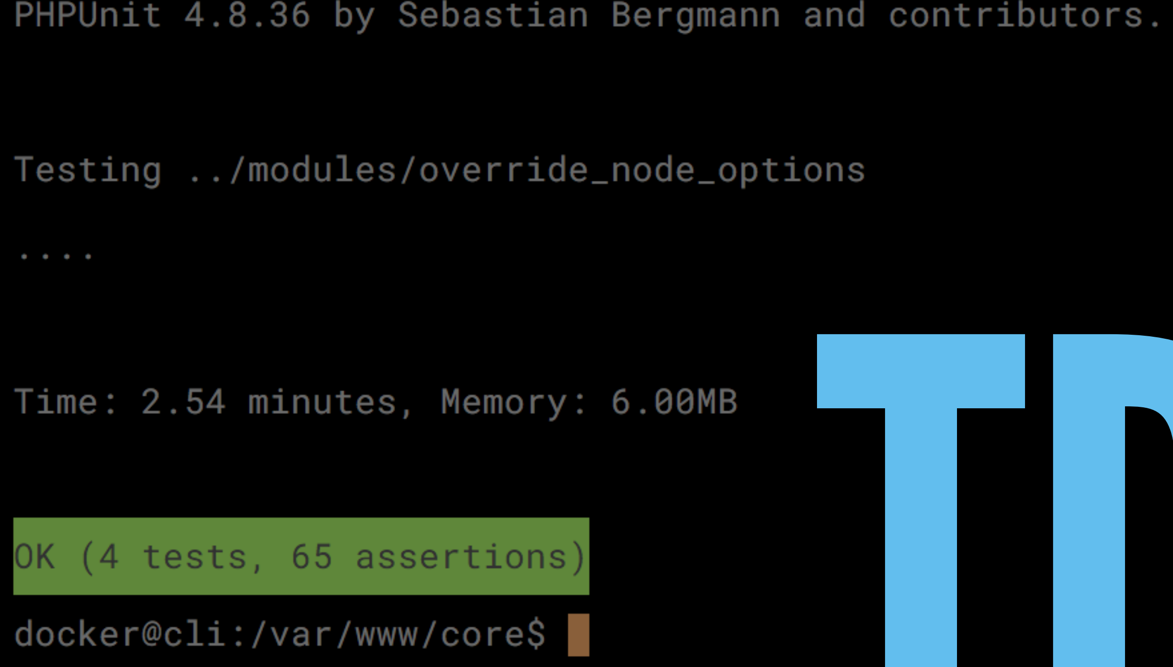


```
PHPUnit 4.8.36 by Sebastian Bergmann and contributors.


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

- Full stack Web Developer & System Administrator
- Senior Developer at Microserve
- Part-time freelancer
- Acquia certified Drupal 8 Grand Master
- Drupal 7 & 8 core contributor
- opdavies (Drupal.org, GitHub, Twitter)
- www.oliverdavies.uk

TESTDRIVENDRUPAL.COM

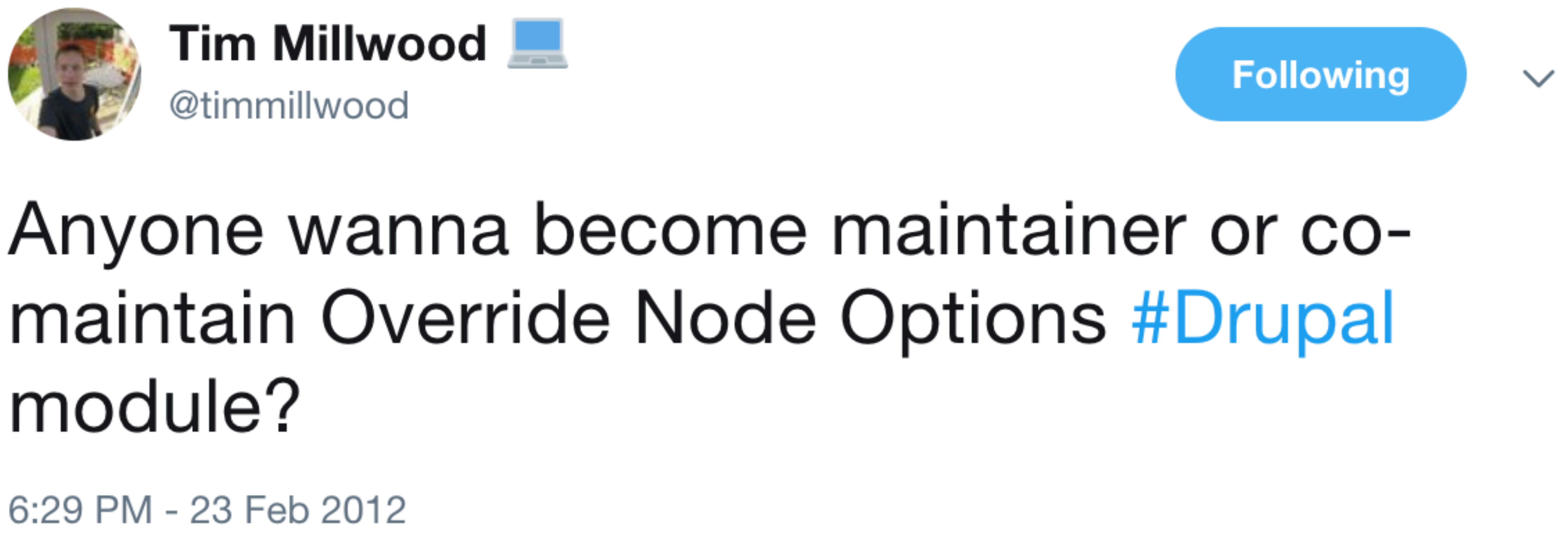
WHY?

WHAT?

HOW?

I WRITE CONTRIB MODULES FOR THE COMMUNITY

I WRITE CUSTOM MODULES FOR CLIENT PROJECTS



Tim Millwood 💻
@timmillwood

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
- First experience of testing with a real module
- 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)
- Crucial to preventing regressions when adding new features or fixing bugs
- Ensured consistency when porting to Drupal 8

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.

https://www.drupal.org/core/gates#testing

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

HOW DO YOU GET QUICKER AT WRITING TESTS?

BY WRITING MORE TESTS

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

SETTING UP YOUR ENVIRONMENT

- Drupal includes `core/phpunit.xml.dist`
- Copy to `core/phpunit.xml`
- Amend values as needed
  - Add base URL, database credentials
- Docksal - `fin addon install phpunit`

WRITING TESTS (DRUPAL 8)

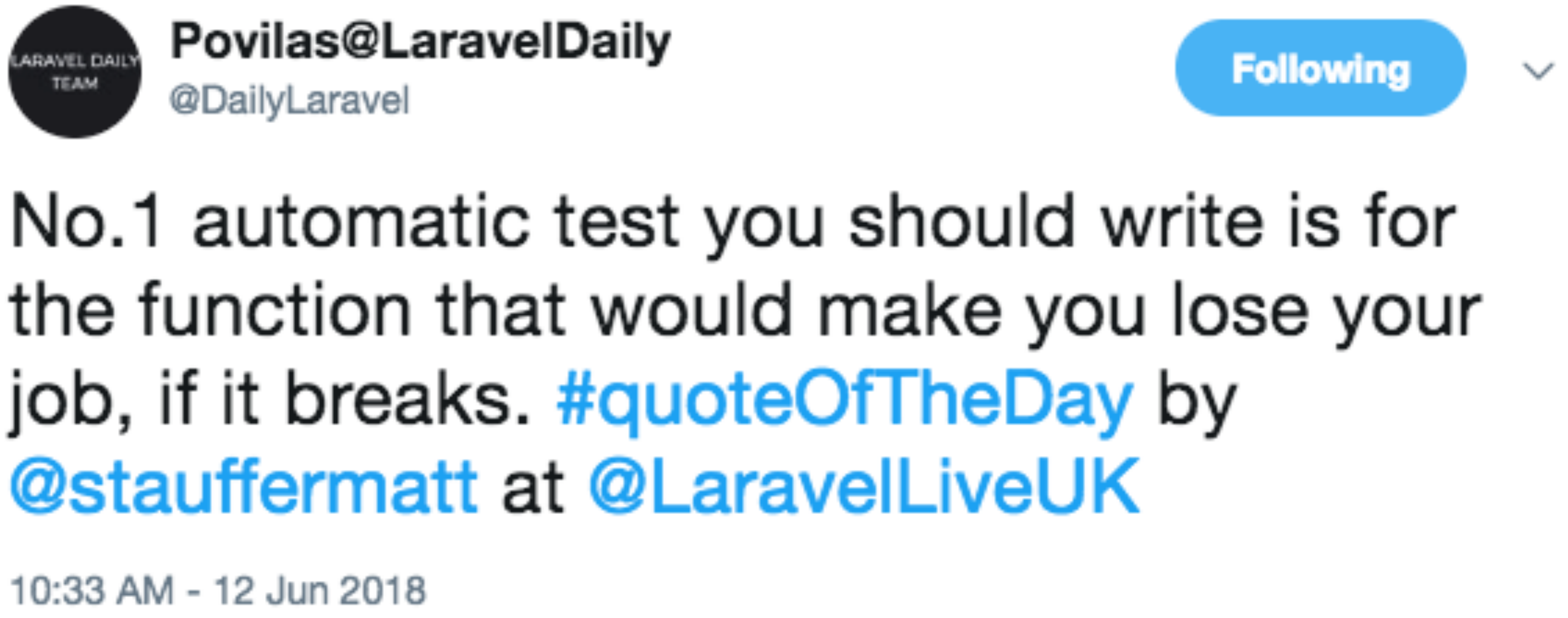
- PHP class with `.php` extension
- `tests/src` directory within each module
- Within the `Drupal\Tests\module_name` namespace
- Class name must match the filename
- Namespace must match the directory structure
- One test class per feature
- Each method must start with `test`

1. ARRANGE
2. ACT
3. ASSERT

```php
<?php

// 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?

- 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

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

TYPES OF TESTS

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

FUNCTIONAL TESTS

- Tests functionality
- Interacts with database
- Full Drupal installation
- Slower to run
- With/without JavaScript

```php
<?php

// modules/phpunit_example/tests/src/Functional/PHPUnitExampleMenuTest.php

namespace Drupal\Tests\phpunit_example\Functional;

use Drupal\Tests\BrowserTestBase;

class PHPUnitExampleMenuTest extends BrowserTestBase {

  public static $modules = ['phpunit_example'];

  public function testPhpUnitExampleMenu() {
    $this->drupalGet('/examples/phpunit-example');

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

```php
<?php

// modules/phpunit_example/tests/src/Functional/PHPUnitExampleMenuTest.php

namespace Drupal\Tests\phpunit_example\Functional;

use Drupal\Tests\BrowserTestBase;

class PHPUnitExampleMenuTest extends BrowserTestBase {

  public static $modules = ['phpunit_example'];

  public function testPhpUnitExampleMenu() {
    $this->drupalGet('/examples/phpunit-example');

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

```php
<?php

// modules/phpunit_example/tests/src/Functional/PHPUnitExampleMenuTest.php

namespace Drupal\Tests\phpunit_example\Functional;

use Drupal\Tests\BrowserTestBase;

class PHPUnitExampleMenuTest extends BrowserTestBase {

  public static $modules = ['phpunit_example'];

  public function testPhpUnitExampleMenu() {
    $this->drupalGet('/examples/phpunit-example');

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

```php
<?php

// modules/phpunit_example/tests/src/Functional/PHPUnitExampleMenuTest.php

namespace Drupal\Tests\phpunit_example\Functional;

use Drupal\Tests\BrowserTestBase;

class PHPUnitExampleMenuTest extends BrowserTestBase {

  public static $modules = ['phpunit_example'];

  public function testPhpUnitExampleMenu() {
    $this->drupalGet('/examples/phpunit-example');

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

```php
<?php

// modules/phpunit_example/tests/src/Functional/PHPUnitExampleMenuTest.php

namespace Drupal\Tests\phpunit_example\Functional;

use Drupal\Tests\BrowserTestBase;

class PHPUnitExampleMenuTest extends BrowserTestBase {

  public static $modules = ['phpunit_example'];

  public function testPhpUnitExampleMenu() {
    $this->drupalGet('/examples/phpunit-example');

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

KERNEL TESTS

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

UNIT TESTS

- Tests PHP logic
- No database interaction
- Fast to run
- Tightly coupled
- Complicated mocking

```php
namespace Drupal\collection_class;

class Collection implements \Countable, \IteratorAggregate {
  private $items;

  public function __construct($items = array()) {
    $this->items = is_array($items) ? $items
      : $this->getArrayableItems($items);
  }

  public function __toString() {
    return $this->toJson();
  }

  public function all() {
    return $this->items;
  }

  public function count() {
    return count($this->items);
  }


  public function isEmpty() {
    return empty($this->items);
  }

  public function first() {
    return array_shift($this->items);
  }
}
```

```php
$collection = new Collection([1, 2, 3, 4, 5]);

// Returns all items.
$collection->all();

// Counts the number of items.
$collection->count();

// Returns the array keys.
$collection->keys();
```

```
public function testAll() {
  $items = ['foo', 'bar', 'baz'];

  $collection = new Collection($items);

  $this->assertEqual($items, $collection->all());
}
```

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

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()

assertNull()
assertNotNull()

assertCount()
assertEmpty()

assertArraySubset()
```

ASSERTIONS (FUNCTIONAL)

```
pageTextContains()
pageTextNotContains()

linkByHrefExists()
linkByHrefNotExists()

statusCodeEquals()
statusCodeNotEquals()
```

REAL LIFE
EXAMPLE

broadbean
technology

SPECIFICATION

- Job adverts created on third-party system, needs to create nodes in Drupal, links users to separate application system
- Adverts need to be linked to offices
- Advert 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

```
$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 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`

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

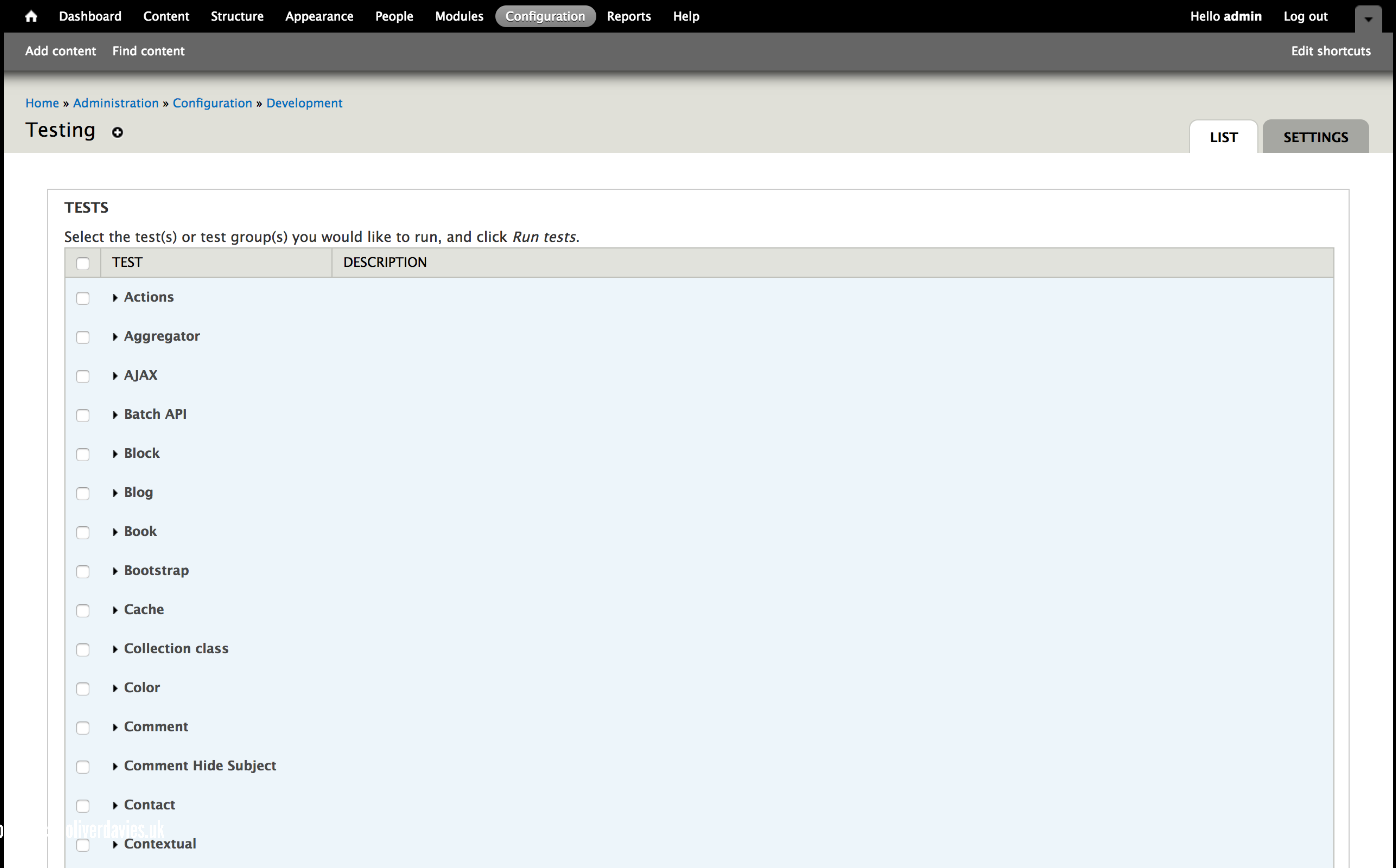
TYPES OF TESTS

- 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

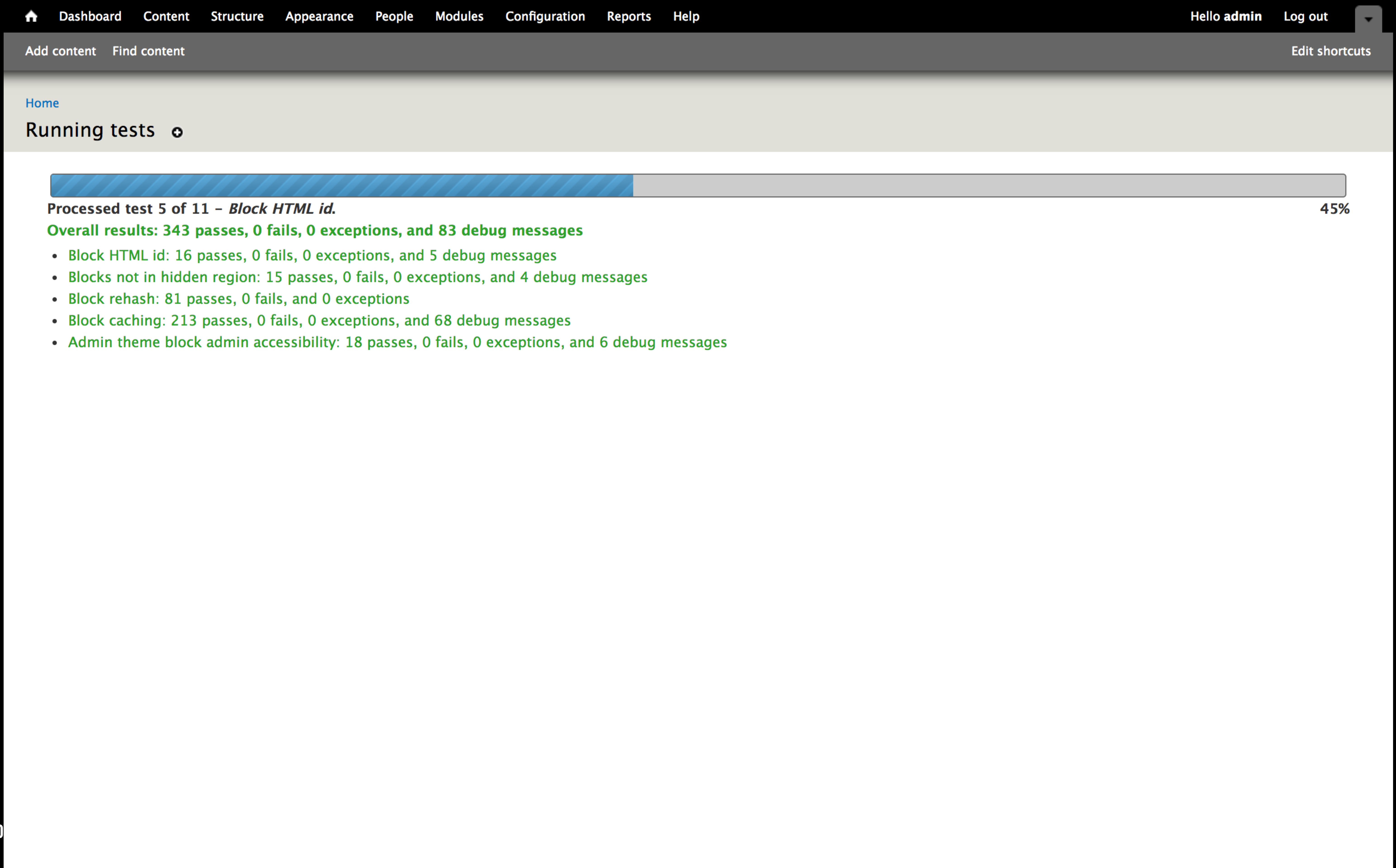


Testing

TESTS

Select the test(s) or test group(s) you would like to run, and click *Run tests*.

□	TEST	DESCRIPTION
□	▸ Actions	
□	▸ Aggregator	
□	▸ AJAX	
□	▸ Batch API	
□	▸ Block	
□	▸ Blog	
□	▸ Book	
□	▸ Bootstrap	
□	▸ Cache	
□	▸ Collection class	
□	▸ Color	
□	▸ Comment	
□	▸ Comment Hide Subject	
□	▸ Contact	
□	▸ Contextual	

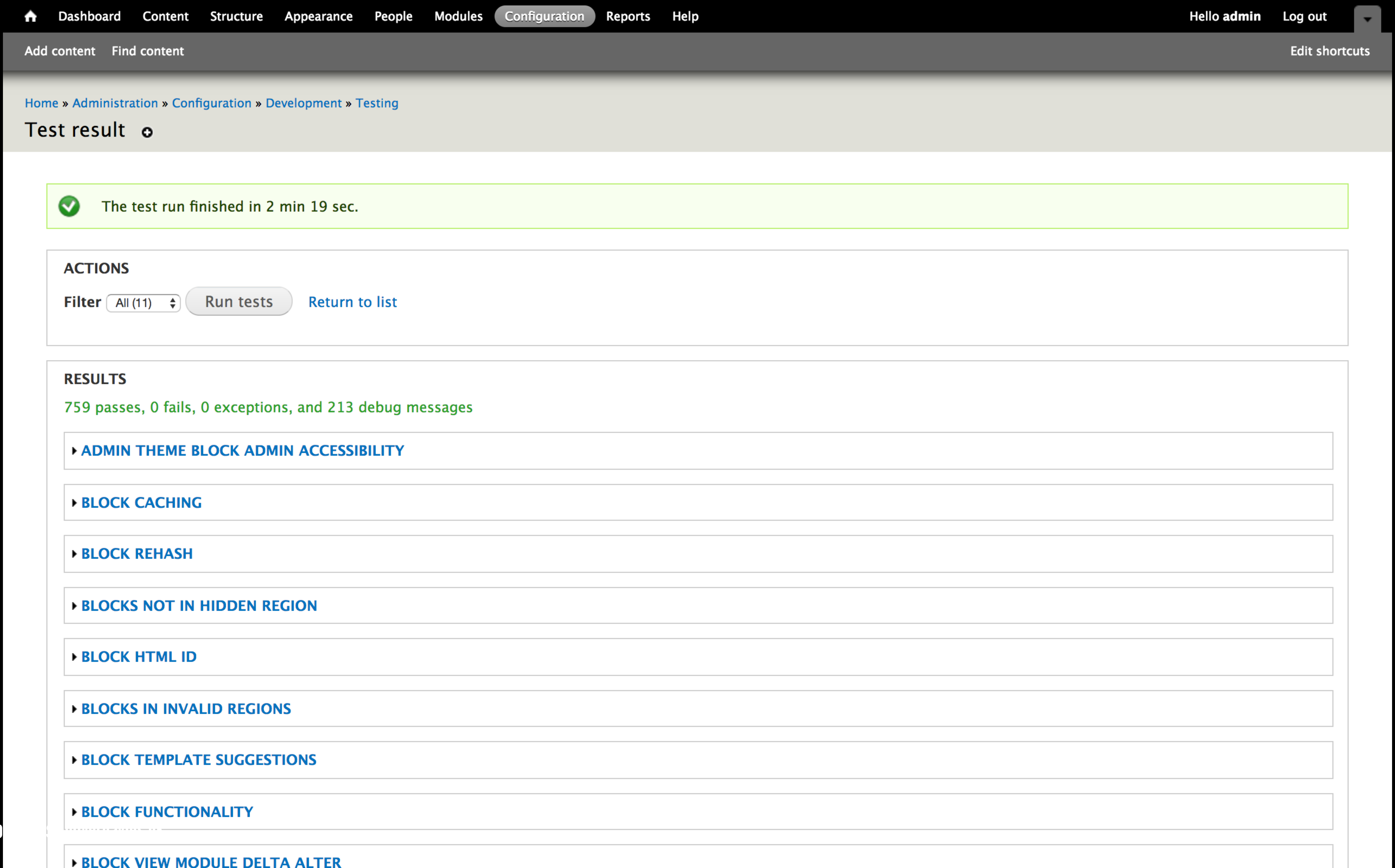


Running tests

Processed test 5 of 11 - *Block HTML id*. **45%**

Overall results: 343 passes, 0 fails, 0 exceptions, and 83 debug messages

- Block HTML id: 16 passes, 0 fails, 0 exceptions, and 5 debug messages
- Blocks not in hidden region: 15 passes, 0 fails, 0 exceptions, and 4 debug messages
- Block rehash: 81 passes, 0 fails, and 0 exceptions
- Block caching: 213 passes, 0 fails, 0 exceptions, and 68 debug messages
- Admin theme block admin accessibility: 18 passes, 0 fails, 0 exceptions, and 6 debug messages



Test result

The test run finished in 2 min 19 sec.

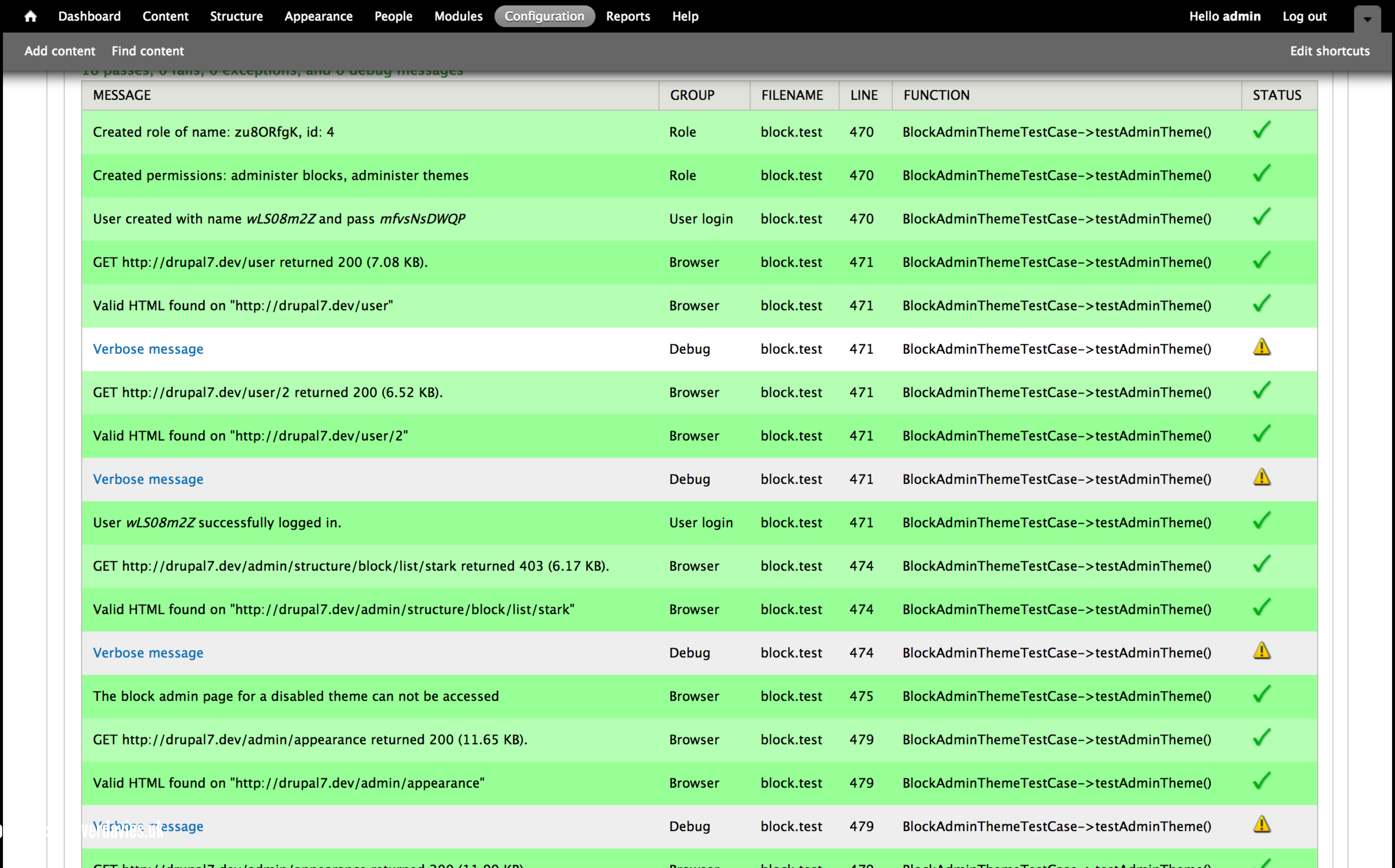
ACTIONS

Filter: All (11) | Run tests | Return to list

RESULTS

759 passes, 0 fails, 0 exceptions, and 213 debug messages

- ADMIN THEME BLOCK ADMIN ACCESSIBILITY
- BLOCK CACHING
- BLOCK REHASH
- BLOCKS NOT IN HIDDEN REGION
- BLOCK HTML ID
- BLOCKS IN INVALID REGIONS
- BLOCK TEMPLATE SUGGESTIONS
- BLOCK FUNCTIONALITY
- BLOCK VIEW MODULE DELTA ALTER



18 passes, 0 fails, 0 exceptions, and 0 debug messages

MESSAGE	GROUP	FILENAME	LINE	FUNCTION	STATUS
Created role of name: zu8ORfgK, id: 4	Role	block.test	470	BlockAdminThemeTestCase->testAdminTheme()	✓
Created permissions: administer blocks, administer themes	Role	block.test	470	BlockAdminThemeTestCase->testAdminTheme()	✓
User created with name *wLS08m2Z* and pass *mfvsNsDWQP*	User login	block.test	470	BlockAdminThemeTestCase->testAdminTheme()	✓
GET http://drupal7.dev/user returned 200 (7.08 KB).	Browser	block.test	471	BlockAdminThemeTestCase->testAdminTheme()	✓
Valid HTML found on "http://drupal7.dev/user"	Browser	block.test	471	BlockAdminThemeTestCase->testAdminTheme()	✓
Verbose message	Debug	block.test	471	BlockAdminThemeTestCase->testAdminTheme()	⚠
GET http://drupal7.dev/user/2 returned 200 (6.52 KB).	Browser	block.test	471	BlockAdminThemeTestCase->testAdminTheme()	✓
Valid HTML found on "http://drupal7.dev/user/2"	Browser	block.test	471	BlockAdminThemeTestCase->testAdminTheme()	✓
Verbose message	Debug	block.test	471	BlockAdminThemeTestCase->testAdminTheme()	⚠
User *wLS08m2Z* successfully logged in.	User login	block.test	471	BlockAdminThemeTestCase->testAdminTheme()	✓
GET http://drupal7.dev/admin/structure/block/list/stark returned 403 (6.17 KB).	Browser	block.test	474	BlockAdminThemeTestCase->testAdminTheme()	✓
Valid HTML found on "http://drupal7.dev/admin/structure/block/list/stark"	Browser	block.test	474	BlockAdminThemeTestCase->testAdminTheme()	✓
Verbose message	Debug	block.test	474	BlockAdminThemeTestCase->testAdminTheme()	⚠
The block admin page for a disabled theme can not be accessed	Browser	block.test	475	BlockAdminThemeTestCase->testAdminTheme()	✓
GET http://drupal7.dev/admin/appearance returned 200 (11.65 KB).	Browser	block.test	479	BlockAdminThemeTestCase->testAdminTheme()	✓
Valid HTML found on "http://drupal7.dev/admin/appearance"	Browser	block.test	479	BlockAdminThemeTestCase->testAdminTheme()	✓
Verbose message	Debug	block.test	479	BlockAdminThemeTestCase->testAdminTheme()	⚠
GET http://drupal7.dev/admin/appearance returned 200 (11.90 KB).	Browser	block.test	479	BlockAdminThemeTestCase->testAdminTheme()	✓

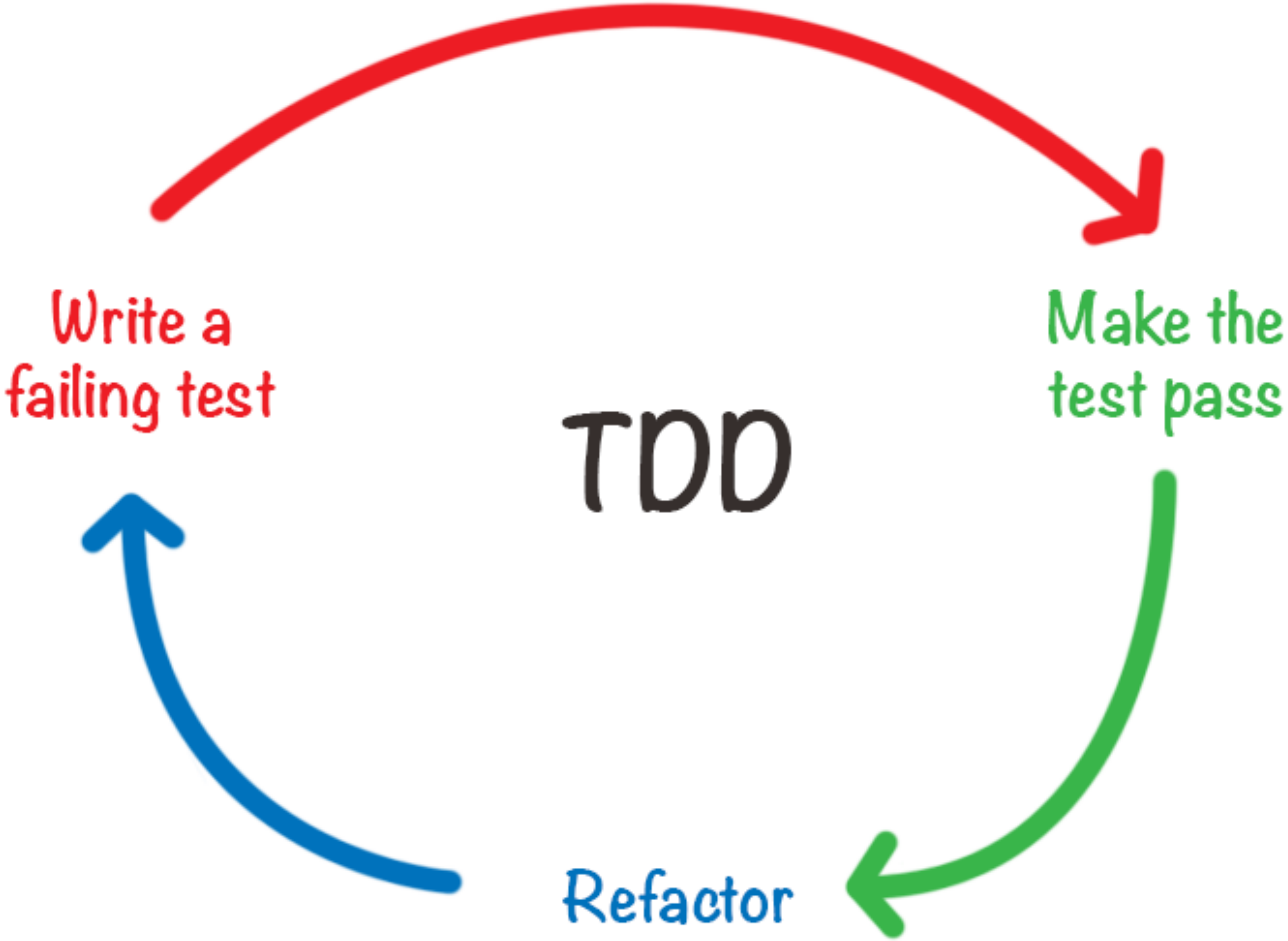
```
$ php core/scripts/run-tests.sh

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

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

```
vendor/bin/phpunit -c core path/to/module

vendor/bin/phpunit -c core path/to/module --filter testSomething

vendor/bin/phpunit -c core path/to/module --verbose
```

TEST DRIVEN DEVELOPMENT

TEST DRIVEN DEVELOPMENT

- Write a test
- See it fail
- Write code until test passes
- Refactor when tests are green
- Repeat



Write a failing test

TDD

Make the test pass

Refactor

https://github.com/foundersandcoders/testing-tdd-intro

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
- Write assertions first, sometimes

BUILDING A NEW DRUPAL 8 MODULE WITH TDD

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 mininum amount at each step, make no assumptions, let the tests guide us
- Start with functional test

```yaml
# tdd_blog.info.yml

name: 'TDD Example'
core: '8.x'
type: 'module'
```

TASK 1:

ENSURE THE BLOG PAGE EXISTS

```php
<?php

// 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);
  }

}
```

```php
<?php

// 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);
  }

}
```

```php
<?php

// 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);
  }

}
```

```php
<?php

// 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);
  }

}
```

```php
<?php

// 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
- Set the path
- Export the config
- Copy it into the module's `config/install` directory

```
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.
```

```yaml
# tdd_blog.info.yml

name: 'TDD Dublin'
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

TASK 2:

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());
}
```

```php
// 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());
}
```

```php
// 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

```php
<?php

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]);
  }

}
```

```php
<?php

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]);
  }

}
```

```php
<?php

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]);
  }

}
```

```php
<?php

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]);
}
```

```php
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 3 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 3 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.
```

- There are no filters on the view
- Add the filters
- Export and save the view

```
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

TASK 3:

ENSURE THE ARTICLES ARE SHOWN IN THE CORRECT ORDER

```
// 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.
}
```

```php
// 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.
}
```

```
// 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.
}
```

```php
// 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.
}
```

```php
// 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 => '3'
+    1 => '2'
+    2 => '4'
+    3 => '1'

/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 => '3'
+    1 => '2'
+    2 => '4'
+    3 => '1'

/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.
```

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

```
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~~

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.

WHEN YOU DO THINGS RIGHT
PEOPLE WON'T BE SURE YOU'VE DONE ANYTHING AT ALL
imgflip.com

RESOURCES

- github.com/opdavies/drupal-module-tdd-dublin
- drupalize.me/series/testing-drupal-7-simpletest
- lullabot.com/articles/an-overview-of-testing-in-drupal-8
- mediacurrent.com/blog/writing-simple-simpletest-tests-your-d7-module
- mediacurrent.com/blog/writing-simple-phpunit-tests-your-d8-module
- knpuniversity.com/screencast/phpunit
- adamwathan.me/test-driven-laravel
- laracasts.com

- oliverdavies.uk/talks
- oliverdavies.uk/twitter
- oliverdavies.uk/drupal
- oliverdavies.uk/github
- oliverdavies.uk/youtube

QUESTIONS?

THANKS