



**DrupalCon**  
GLOBAL 2020  
JULY 14-17

# Introducing Drupal Test Traits

Moshe Weitzman  
Independent Consultant



# Moshe Weitzman

Independant Consultant



- Core contributor since 2001
- Drush and Devel maintainer
- Started groups.drupal.org, DCOC, #D7CX, DrupalSpoons, DTT
- Drupal Security Team



# Agenda

- Introduction
- When to use DTT (and when to use Behat)
- Install and run tests
- Review three example test cases
- Review DrupalTrait
- Add-ons
- Open Discussion



“

Traits for testing Drupal sites that  
have user content (versus  
unpopulated sites).

”

DTT's README.md



“



Behat is like no other test framework.  
Instead of proving that your system  
was built right, it helps build the right  
system in the first place by facilitating  
& enriching requirements  
communication. ”

[docs.behat.org](https://docs.behat.org)



## Install and run

1. `composer require weitzman/drupal-test-traits`
2. `DTT_BASE_URL=http://example.com`
3. `phpunit --bootstrap=/bootstrap-fast.php  
web/modules/custom`



# Three Test Cases



```
class SimpleExampleTest extends ExistingSiteBase {
  public function testArticleContentType() {
    // Create a user+term+node. Will be automatically cleaned up.
    $author = $this->createUser([], null, true);
    $vocab = Vocabulary::Load('tags');
    $term = $this->createTerm($vocab);
    $node = $this->createNode([
      'title' => 'Llama',
      'type' => 'article',
      'field_tags' => ['target_id' => $term->id()],
      'uid' => $author->id(),
    ]);
    $node->setPublished()->save();

    // We can browse pages.
    $this->drupalGet($node->toUrl());
    $this->assertSession()->statusCodeEquals(200);

    // We can login and browse admin pages.
    $this->drupalLogin($author);
    $this->drupalGet($node->toUrl('edit-form'));
  }
}
```





```
class ExampleMailTest extends ExistingSiteBase {
```

```
  use MailCollectionAssertTrait;  
  use MailCollectionTrait;  
  use QueueRunnerTrait;
```

```
  protected function setUp() {  
    $this->startMailCollection();  
    $this->clearQueue('mass_unpublish_reminders_queue');  
  }
```

```
  public function testUnpublishReminder() {
```


```
    $this->createNode([  
      'uid' => $this->author->id(),  
      'unpublish_on' => strtotime('+3 days', \Drupal::time()->getRequestTime()),  
    ]);  
    // Send the emails.  
    $this->runQueue('mass_unpublish_reminders_queue');
```

```
    $this->assertMailCollection()  
      ->seekToModule('mass_unpublish')  
      ->seekToRecipient($this->author->mail->value)  
      ->countEquals(1);
```

```
    $mails = $this->getMails();  
    $this->assertTrue(isset($mails[0]['headers']['cc']));  
  }
```



```
class ExampleSelenium2DriverTest extends ExistingSiteSelenium2DriverTestBase {  
    public function testContentCreation() {  
        // Test autocomplete on article creation.  
        $web_assert = $this->assertSession();  
        $this->drupalGet('/node/add/article');  
        $page = $this->getCurrentPage();  
        $page->fillField('title[0][value]', 'Article Title');  
        $tags = $page->findField('field_tags[target_id]');  
        $tags->setValue('Ter');  
        $tags->keyDown('m');  
        $result = $web_assert->waitForElementVisible('css', '.ui-autocomplete li');  
        $this->assertNotNull($result);  
        // Click the autocomplete option  
        $result->click();  
        $web_assert->pageTextContains('Term 1');  
        $submit_button = $page->findButton('Save');  
        $submit_button->press();  
  
        // Verify the term on the page.  
        $web_assert->pageTextContains('Article Title');  
        $web_assert->pageTextContains('Term 1');  
    }  
}
```



# The DrupalTrait



```
trait DrupalTrait {
```

```
// Bootstrap Drupal. Call this from your setUp() method.
```

```
public function setupDrupal() {  
    $finder = new DrupalFinder();  
    $finder->locateRoot(getcwd());  
    $classLoader = include $finder->getVendorDir() . '/autoload.php';
```

```
    $server = [  
        'HTTP_HOST' => $host,  
        'SERVER_PORT' => $port,  
        'REQUEST_URI' => $path . '/',  
        'SCRIPT_FILENAME' => $path . '/index.php',  
    ];
```

```
    $request = Request::create($this->baseUri . '/', 'GET', [], [], [], $server);  
    $this->kernel = DrupalKernel::createFromRequest(  
        $request,  
        $classLoader,  
        'existing-site-testcase',  
        false,  
        $finder->getDrupalRoot()  
    );
```

```
    chdir($finder->getDrupalRoot());  
    $this->kernel->boot();  
    $this->kernel->preHandle($request);  
    $this->container = $this->kernel->getContainer();
```

```
}
```





## More Traits

- MailCollectionTrait
- ScreenShotTrait
- LoginTrait (contrib)
- QueueRunnerTrait (contrib)



# DrupalCon GLOBAL

The Open Source Digital Experience Conference

**ONLINE EVERYWHERE**

**2020 JULY 14-17**