Drupal Drivers Documentation

Release 1.0

Jonathan Hedstrom

Nov 22, 2019

Contents

1 Installation 3

2 Comparison of Drivers 5

3 Usage 7

The Drupal Drivers are a collection of light-weight drivers with a common interface for interacting with Drupal. These are generally intended for testing, and are not meant to be API-complete.

Contents:

CHAPTER 1

Installation

To utilize the Drupal Drivers in your own project, they are installed via composer.

```
{
  "require": {
    "drupal/drupal-driver": "~1.0"
  }
}
```

and then install and run composer

```
$> curl -sS http://getcomposer.org/installer | php
$> php composer.phar install
```

If you plan on using the Drush driver, then you need to ensure that the behat-drush-endpoint is available in the target Drupal site. There are two ways to do this:

1. Copy the files manually. The project can be found at:

https://github.com/drush-ops/behat-drush-endpoint

2. Use Composer.

If you are using Composer to manage your Drupal site, then you only need to require drupal/drupal-driver and composer/installers, and the behat-drush-endpoint files will be copied to the correct location.

CHAPTER 2

Comparison of Drivers

The available drivers for interacting with your site, which are compatible with Drupal 7, and 8. Each driver has its own limitiations.

Feature	Blackbox	Drush	Drupal API
Create users	No	Yes	Yes
Create nodes	No	[*]	Yes
Create vocabularies	No	No	Yes
Create taxonomy terms	No	[*]	Yes
Run tests and site on different servers	Yes	Yes	No

[*] Possible if behat.d7.drush.inc or behat.d8.drush.inc, as appropriate, is installed in the target Drupal site.

CHAPTER 3

Usage

3.1 Drupal API driver

```php
<?php

use Drupal\Driver\DrupalDriver;
use Drupal\Driver\Cores\Drupal8;

require 'vendor/autoload.php';

// Path to Drupal.
$path = './drupal-8';

// Host.
$uri = 'http://d8.devl';

$driver = new DrupalDriver($path, $uri);
$driver->setCoreFromVersion();

// Bootstrap Drupal.
$driver->bootstrap();

// Create a node.
$node = (object) array(
  'type' => 'article',
  'uid' => 1,
  'title' => $driver->getRandom()->name(),
);
$driver->createNode($node);
```

3.2 Drush driver

```php
<?php

use Drupal\Driver\DrushDriver;

...

$alias = '@mysite';
$driver = new DrushDriver($alias);

...
```

3.3 Blackbox

Note, the blackbox driver has no ability to control Drupal, and is provided as a fallback for when some tests can run without such access.

Any testing application should catch unsupported driver exceptions.

```php
<?php

use Drupal\Driver\BlackboxDriver;
use Drupal\Driver\Exception\UnsupportedDriverActionException;

...

$driver = new BlackboxDriver($alias);

try {
  // Create a node.
  $node = (object) array(
    'type' => 'article',
    'uid' => 1,
    'title' => $driver->getRandom()->name(),
  );
  $driver->createNode($node);
}
catch (UnsupportedDriverActionException $e) {
  // Mark test as skipped.
}
```

3.4 Practical example with PHPUnit

By using the phpunit/mink project in conjunction with the Drupal Driver, one can use PHPUnit to drive browser sessions and control Drupal.

To install:

```json
{
    "require": {
        "aik099/phpunit-mink": "~2.0",
```

(continues on next page)

(continued from previous page)

```
        "drupal/drupal-driver": "~1.0"
    }
}
```

and then, in the tests directory, a sample test:

```
<?php

use aik099\PHPUnit\BrowserTestCase;

use Drupal\Driver\DrupalDriver;

class GeneralTest extends BrowserTestCase
{

    /**
     * @var \Drupal\Driver\DriverInterface
     */
     protected static $driver;

    // Path to a Drupal install. This example assumes the directory is in the same␣
↪one as the `composer.json` file.
    protected static $drupalRoot = './drupal';

    // Url to the homepage of the Drupal install.
    protected static $uri = 'http://d8.devl';

    public static $browsers = array(
        // Selenium info.
        array(
            'host' => 'localhost',
            'port' => 4444,
            'browserName' => 'firefox',
            'baseUrl' => 'http://d8.devl',
        ),
    );

     public static function setUpBeforeClass() {
        self::$driver = new DrupalDriver(static::$drupalRoot, static::$uri);
        self::$driver->setCoreFromVersion();
        self::$driver->bootstrap();
    }

    public function testUsingSession()
    {
        // This is Mink's Session.
        $session = $this->getSession();

        // Go to a page.
        $session->visit(static::$uri);

        // Validate text presence on a page.
        $this->assertTrue($session->getPage()->hasContent('Site-Install'));
    }

    public function testUsingBrowser()
    {
```

(continues on next page)

(continued from previous page)

```
        // Prints the name of used browser.
        echo sprintf(
            "I'm executed using '%s' browser",
            $this->getBrowser()->getBrowserName()
        );
    }

    public function testNodeCreate() {
        $drupal = self::$driver;
        $node = (object) [
            'title' => $drupal->getRandom()->string(),
            'type' => 'article',
        ];
        $drupal->createNode($node);

        $session = $this->getSession();
        $session->visit(static::$uri . '/node/' . $node->nid);

        $this->assertTrue($session->getPage()->hasContent($node->title));
    }

}
```