

---

# **Drupal Drivers Documentation**

*Release 1.0*

**Jonathan Hedstrom**

**Nov 22, 2019**



---

## Contents

---

<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Comparison of Drivers</b>	<b>5</b>
<b>3</b>	<b>Usage</b>	<b>7</b>



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.





---

### Comparison of Drivers

---

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

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.



### 3.1 Drupal API driver

```
1 <?php
2
3 use Drupal\Driver\DrupalDriver;
4 use Drupal\Driver\Cores\Drupal8;
5
6 require 'vendor/autoload.php';
7
8 // Path to Drupal.
9 $path = './drupal-8';
10
11 // Host.
12 $uri = 'http://d8.devl';
13
14 $driver = new DrupalDriver($path, $uri);
15 $driver->setCoreFromVersion();
16
17 // Bootstrap Drupal.
18 $driver->bootstrap();
19
20 // Create a node.
21 $node = (object) array(
22     'type' => 'article',
23     'uid' => 1,
24     'title' => $driver->getRandom()->name(),
25 );
26 $driver->createNode($node);
```

## 3.2 Drush driver

```
1 <?php
2
3 use Drupal\Driver\DrushDriver;
4
5 ...
6
7 $alias = '@mysite';
8 $driver = new DrushDriver($alias);
9
10 ...
```

## 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.

```
1 <?php
2
3 use Drupal\Driver\BlackboxDriver;
4 use Drupal\Driver\Exception\UnsupportedDriverActionException;
5
6 ...
7
8 $driver = new BlackboxDriver($alias);
9
10 try {
11     // Create a node.
12     $node = (object) array(
13         'type' => 'article',
14         'uid' => 1,
15         'title' => $driver->getRandom()->name(),
16     );
17     $driver->createNode($node);
18 }
19 catch (UnsupportedDriverActionException $e) {
20     // Mark test as skipped.
21 }
```

## 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:

```
1 {
2     "require": {
3         "aik099/phpunit-mink": "~2.0",
```

(continues on next page)

(continued from previous page)

```

4     "drupal/drupal-driver": "~1.0"
5   }
6 }

```

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

```

1 <?php
2
3 use aik099\PHPUnit\BrowserTestCase;
4
5 use Drupal\Driver\DrupalDriver;
6
7 class GeneralTest extends BrowserTestCase
8 {
9
10    /**
11     * @var \Drupal\Driver\DriverInterface
12     */
13    protected static $driver;
14
15    // Path to a Drupal install. This example assumes the directory is in the same_
16    ↪one as the `composer.json` file.
17    protected static $drupalRoot = './drupal';
18
19    // Url to the homepage of the Drupal install.
20    protected static $uri = 'http://d8.dev1';
21
22    public static $browsers = array(
23        // Selenium info.
24        array(
25            'host' => 'localhost',
26            'port' => 4444,
27            'browserName' => 'firefox',
28            'baseUrl' => 'http://d8.dev1',
29        ),
30    );
31
32    public static function setUpBeforeClass() {
33        self::$driver = new DrupalDriver(static::$drupalRoot, static::$uri);
34        self::$driver->setCoreFromVersion();
35        self::$driver->bootstrap();
36    }
37
38    public function testUsingSession()
39    {
40        // This is Mink's Session.
41        $session = $this->getSession();
42
43        // Go to a page.
44        $session->visit(static::$uri);
45
46        // Validate text presence on a page.
47        $this->assertTrue($session->getPage()->hasContent('Site-Install'));
48    }
49
50    public function testUsingBrowser()
51    {

```

(continues on next page)

(continued from previous page)

```
51     // Prints the name of used browser.
52     echo sprintf(
53         "I'm executed using '%s' browser",
54         $this->getBrowser()->getBrowserName()
55     );
56 }
57
58 public function testNodeCreate() {
59     $drupal = self::$driver;
60     $node = (object) [
61         'title' => $drupal->getRandom()->string(),
62         'type' => 'article',
63     ];
64     $drupal->createNode($node);
65
66     $session = $this->getSession();
67     $session->visit(static::$uri . '/node/' . $node->nid);
68
69     $this->assertTrue($session->getPage()->hasContent($node->title));
70 }
71
72 }
```