

---

# **Drupal Drivers Documentation**

*Release 1.0*

**Jonathan Hedstrom**

April 08, 2016



<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:



---

## 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
```





---

## 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	No	Yes
Create vocabularies	No	No	Yes
Create taxonomy terms	No	No	Yes
Run tests and site on different servers	Yes	Yes	No



### 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 phunit/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",
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 $driver;
14
15 // Path to a Drupal install. This example assumes the directory is in the same one as the `composer
16 protected static $drupalRoot = './drupal';
17
18 // Url to the homepage of the Drupal install.
19 protected static $uri = 'http://d8.dev1';
20
21 public static $browsers = array(
22     // Selenium info.
23     array(
24         'host' => 'localhost',
25         'port' => 4444,
26         'browserName' => 'firefox',
27         'baseUrl' => 'http://d8.dev1',
28     ),
29 );
30
31 public function setUp() {
32     $this->driver = new DrupalDriver(static::$drupalRoot, static::$uri);
33     $this->driver->setCoreFromVersion();
34     $this->driver->bootstrap();
35 }
36
37 public function testUsingSession()
38 {
39     // This is Mink's Session.
40     $session = $this->getSession();
41
42     // Go to a page.
43     $session->visit(static::$uri);
44
45     // Validate text presence on a page.
46     $this->assertTrue($session->getPage()->hasContent('Site-Install'));
47 }
48
49 public function testUsingBrowser()
50 {
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     $node = (object) [
60         'title' => $this->driver->getRandom()->string(),
61         'type' => 'article',
62     ];
63     $this->driver->createNode($node);
64
65     $session = $this->getSession();

```

```
66     $session->visit(static::$Suri . '/node/' . $node->nid);
67
68     $this->assertTrue($session->getPage()->hasContent($node->title));
69 }
70
71 }
```