

Prologue

Getting Started

Architecture Concepts

The Basics

Frontend

Security

Digging Deeper

Database

Eloquent ORM

Testing

• Getting Started

HTTP Tests

Console Tests

Browser Tests

Database

Mocking

Official Packages



Deploy Laravel and PHP applications on DigitalOcean, Linode, & AWS.

ADS VIA CARBON

Testing: Getting Started

Introduction

Environment

Creating & Running Tests

Introduction

Laravel is built with testing in mind. In fact, support for testing with PHPUnit is included out of the box and a `phpunit.xml` file is already set up for your application. The framework also ships with convenient helper methods that allow you to expressively test your applications.

By default, your application's `tests` directory contains two directories: `Feature` and `Unit`. Unit tests are tests that focus on a very small, isolated portion of your code. In fact, most unit tests probably focus on a single method. Feature tests may test a larger portion of your code, including how several objects interact with each other or even a full HTTP request to a JSON endpoint.

An `ExampleTest.php` file is provided in both the `Feature` and `Unit` test directories. After installing a new Laravel application, run `phpunit` on the command line to run your tests.

Environment

When running tests via `phpunit`, Laravel will automatically set the configuration environment to `testing` because of the environment variables defined in the `phpunit.xml` file. Laravel also automatically configures the session and cache to the `array` driver while testing, meaning no session or cache data will be persisted while testing.

You are free to define other testing environment configuration values as necessary. The `testing` environment variables may be configured in the `phpunit.xml` file, but make sure to clear your configuration cache using the `config:clear` Artisan command before running your tests!

In addition, you may create a `.env.testing` file in the root of your project. This file will override the `.env` file when running PHPUnit tests or executing Artisan commands with the `--env=testing` option.

Creating & Running Tests

To create a new test case, use the `make:test` Artisan command:

```
// Create a test in the Feature directory...
php artisan make:test UserTest

// Create a test in the Unit directory...
php artisan make:test UserTest --unit
```

Once the test has been generated, you may define test methods as you normally would using PHPUnit. To run your tests, execute the `phpunit` command from your terminal:

```
<?php

namespace Tests\Unit;

use Illuminate\Foundation\Testing\RefreshDatabase;
use Tests\TestCase;

class ExampleTest extends TestCase
{
    /**
     * A basic test example.
     *
     * @return void
     */
    public function testBasicTest()
    {
        $this->assertTrue(true);
    }
}
```



If you define your own `setUp` / `tearDown` methods within a test class, be sure to call the respective `parent::setUp()` / `parent::tearDown()` methods on the parent class.

Become a Laravel Partner

Laravel Partners are elite shops providing top-notch Laravel development and consulting. Each of our partners can help you craft a beautiful, well-architected project.

[Our Partners](#)

Laravel

[Release Notes](#)[Getting Started](#)[Routing](#)[Blade Templates](#)[Authentication](#)[Authorization](#)[Artisan Console](#)[Database](#)[Eloquent ORM](#)[Testing](#)[Laracasts](#)[Laravel News](#)[Laracore](#)[Laracore EU](#)[Laracore AU](#)[Jobs](#)[Certification](#)[Forums](#)[Vehikl](#)[Tighten Co.](#)[Kirschbaum](#)[Byte 5](#)[64Robots](#)[Cubet](#)[DevSquad](#)[Ideil](#)[Cyber-Duck](#)[ABOUT YOU](#)[Become A Partner](#)[Vapor](#)[Forge](#)[Envoyer](#)[Horizon](#)[Lumen](#)[Nova](#)[Echo](#)[Valet](#)[Mix](#)[Spark](#)[Cashier](#)[Homestead](#)[Dusk](#)[Passport](#)[Scout](#)[Socialite](#)

elegant syntax. We believe development must be an enjoyable and creative experience to be truly fulfilling. Laravel attempts to take the pain out of development by easing common tasks used in most web projects.

Laravel is a Trademark of Taylor Otwell.
Copyright © 2011-2019 Laravel LLC.

