



# Laravel

## *Laravel Packages & Testing*

*Marco Monteiro*



# Contributors

---

2

► Author(s):

- Marco Monteiro ([marco.monteiro@ipleiria.pt](mailto:marco.monteiro@ipleiria.pt))



# Summary

---

3

1. Packages
2. Testing
3. References



# 1 – PACKAGES



# Packages

---

5

- ▶ We can install third-party packages to add new functionalities to Laravel framework (or create our own packages)
- ▶ Repository of packages for Laravel:

**`https://packalyst.com/`**

- ▶ Repository of php packages:  
`https://packagist.org/`



# Packages

---

6

- ▶ Example of packages:
  - ▶ **Lavachart** – to draw charts
  - ▶ **Excel** – Import / export data to Excel
  - ▶ **laravel-dompdf** – to create PDF documents
  - ▶ **Socialite** - authentication with Facebook, Twitter, Google, LinkedIn, GitHub and Bitbucket
  - ▶ **Spatie/laravel-permission** and **Bouncer** - authorization with roles
  - ▶ **MongoDB** – MongoDB (NoSQL Database) model and queries



## 2 – TESTING



# Testing

8

- ▶ Laravel supports testing with PHPUnit out of the box
  - ▶ Already includes a configuration file (phpunit.xml) set up for your application
  - ▶ Default test directory: /tests
  - ▶ Includes 2 types of tests:
    - ▶ /tests/Unit – **Unit tests** - tests that focus on a very small, isolated portion of your code
    - ▶ /tests/Feature – **Feature tests** - 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.
- ▶ To run the tests, execute phpunit on the Laravel project folder

```
phpunit
```





- ▶ Laravel provides a fluent API for making HTTP requests to your application and examining the output. Example

```
namespace Tests\Feature;

use Tests\TestCase;
. . .

class ExampleTest extends TestCase
{
    public function testBasicTest()
    {
        $response = $this->get('/someurl');

        $response->assertStatus(200);
    }
}
```



- ▶ Laravel provides a variety of custom assertion methods for your PHPUnit tests

[assertCookie](#)

[assertCookieExpired](#)

[assertCookieMissing](#)

[assertDontSee](#)

[assertDontSeeText](#)

[assertExactJson](#)

[assertForbidden](#)

[assertHeader](#)

[assertHeaderMissing](#)

[assertJson](#)

[assertJsonFragment](#)

[assertJsonMissing](#)

[assertJsonMissingExact](#)

[assertJsonStructure](#)

[assertJsonValidationErrors](#)

[assertNotFound](#)

[assertPlainCookie](#)

[assertRedirect](#)

[assertSee](#)

[assertSeeInOrder](#)

[assertSeeText](#)

[assertSeeTextInOrder](#)

[assertSessionHas](#)

[assertSessionHasAll](#)

[assertSessionHasErrors](#)

[assertSessionHasErrorsIn](#)

[assertSessionMissing](#)

[assertStatus](#)

[assertSuccessful](#)

[assertViewHas](#)

[assertViewHasAll](#)

[assertViewsIs](#)

[assertViewMissing](#)



- ▶ Laravel Dusk provides an expressive, easy-to-use browser automation and testing API – simulate the application usage at the browser level.
- ▶ Dusk provides a variety of methods for interacting with HTML elements, forms, input elements, etc...
- ▶ Examples:

```
$browser->click('.selector');  
  
$browser->type('email', 'taylor@laravel.com');  
  
$browser->select('size', 'Large');  
$browser->select('size');           // Random value
```



# Browser Tests

12

- Dusk provides a variety of assertions that you may make against your application

[assertTitle](#)

[assertTitleContains](#)

[assertUrlls](#)

[assertPathBeginsWith](#)

[assertPathIs](#)

[assertPathIsNot](#)

[assertRoutels](#)

[assertQueryStringHas](#)

[assertQueryStringMissing](#)

[assertFragmentIs](#)

[assertFragmentBeginsWith](#)

[assertFragmentIsNot](#)

[assertHasCookie](#)

[assertCookieMissing](#)

[assertCookieValue](#)

[assertPlainCookieValue](#)

[assertSee](#)

[assertDontSee](#)

[assertSeeIn](#)

[assertDontSeeIn](#)

[assertSourceHas](#)

[assertSourceMissing](#)

[assertSeeLink](#)

[assertDontSeeLink](#)

[assertInputValue](#)

[assertInputValueIsNot](#)

[assertChecked](#)

[assertNotChecked](#)

[assertRadioSelected](#)

[assertRadioNotSelected](#)

[assertSelected](#)

[assertNotSelected](#)

[assertSelectHasOptions](#)

[assertSelectMissingOptions](#)

[assertSelectHasOption](#)

[assertValue](#)

[assertVisible](#)

[assertPresent](#)

[assertMissing](#)

[assertDialogOpened](#)

[assertEnabled](#)

[assertDisabled](#)

[assertFocused](#)

[assertNotFocused](#)

[assertVue](#)

[assertVueIsNot](#)

[assertVueContains](#)

[assertVueDoesNotContain](#)



# Database Tests

13

- ▶ Tests that verify if data exists in the database

- ▶ Example:

```
$this->assertDatabaseHas('users', [  
    'email' => 'sally@example.com'  
]);
```

- ▶ Database related asserts

Method	Description
<code>\$this-&gt;assertDatabaseHas(\$table, array \$data);</code>	Assert that a table in the database contains the given data.
<code>\$this-&gt;assertDatabaseMissing(\$table, array \$data);</code>	Assert that a table in the database does not contain the given data.
<code>\$this-&gt;assertSoftDeleted(\$table, array \$data);</code>	Assert that the given record has been soft deleted.



## 3 – REFERENCES



# References

---

15

- ▶ Official Documentation
  - ▶ <https://laravel.com/docs>
  - ▶ <https://laravel.com/docs/testing>
  - ▶ <https://laravel.com/docs/packages>
- ▶ Packalist – packages for Laravel
  - ▶ <https://packalyst.com/>
- ▶ Packagist – php packages
  - ▶ <https://packagist.org/>