

Testing in Laravel

an introduction

About me

René Kulik

PHP Developer at the ABOUT YOU Shop Unit

[@kulik io](#)

<http://www.kulik.io/>

What is software testing?

- Process used to verify the correctness, completeness, security and quality of a software
- Positive and negative conditions
- Intent of finding errors
- Different approaches

Why don't developers test?

- They do not know how to write tests
- They do not feel responsible
- They say writing tests takes too much time
- They are too convinced of their code
- They were instructed not to write tests

Why developers should write tests?

- Reduce bugs in new and old features
- Allow refactoring
- Good documentation
- **Maintain software quality**

Laravel's test setup

- PHPUnit
- Pre-configured phpunit.xml
- Helper methods
- tests-directory
- Feature- & Unit-directory to classify tests

Configure testing environment

phpunit.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<phpunit ... >
  <testsuites> ... </testsuites>
  <filter> ... </filter>
  <php>
    <env name="APP_ENV" value="testing"/>
    <env name="CACHE_DRIVER" value="array"/>
    <env name="SESSION_DRIVER" value="array"/>
    <env name="QUEUE_DRIVER" value="sync"/>
  </php>
</phpunit>
```

Create tests

```
// Create feature test
```

```
~ $ php artisan make:test ExampleTest
```

```
// Create unit test
```

```
~ $ php artisan make:test ExampleTest --unit
```


Example test

```
class ExampleTest extends TestCase
{
    public function testBasicTest()
    {
        $response = $this->get('/');
        $response->assertStatus(200);
    }
}
```

Run tests

```
// All tests
```

```
~ $ phpunit
```

```
// Filter tests
```

```
~ $ phpunit --filter ExampleTest
```

```
// Specific group
```

```
~ $ phpunit --group ExampleGroup
```

Database Testing

Resetting database

```
use DatabaseMigrations;
```

```
use DatabaseTransactions;
```

```
// New in 5.5
```

```
use RefreshDatabase;
```

Database Testing

Seeding database

```
$factory->define(  
    App\Task::class,  
    function (Faker\Generator $faker) {  
        return ['title' => $faker->sentence(5)];  
    }  
);
```

Database Testing

Available assertions

```
$this->assertDatabaseHas($table, array $data);
```

```
$this->assertDatabaseMissing($table, array $data);
```

```
$this->assertSoftDeleted($table, array $data);
```

Database Testing

Example test

```
use DatabaseMigrations;
```

```
public function testDatabase()
```

```
{
```

```
    factory(App\Task::class)->create(['title' => 'Prepare talk']);
```

```
    $this->assertDatabaseHas('tasks', ['title', 'Prepare talk']);
```

```
}
```

Testing APIs

Call given URI with a JSON request

```
public function json(  
    $method,  
    $uri,  
    array $data = [],  
    array $headers = []  
) { ... }
```

Testing APIs

Available assertions

```
$response->assertJson(array $data);
```

```
$response->assertJsonFragment(array $data);
```

```
$response->assertJsonMissing(array $data);
```

```
$response->assertExactJson(array $data);
```

```
$response->assertJsonStructure(array $structure);
```


Testing APIs

Example test

```
public function testCreateTask()
{
    $this->json('POST', '/tasks', ['title' => 'Prepare talk'])
        ->assertStatus(201);
}
```

Browser tests

- Laravel Dusk
- Uses standalone ChromeDriver installation
- Neither JDK nor Selenium required
- New in 5.5: headless by default

Browser tests

Example test

```
public function testLoginInBrowser()
{
    $user = factory(User::class)->create(['email' => 'example@example.com']);
    $this->browse(function ($browser) use ($user) {
        $browser->visit('/login')
            ->type('email', $user->email)
            ->type('password', 'password123')
            ->press('Login')
            ->assertPathIs('/youAreLoggedIn');
    });
}
```

Exception Handling

New in 5.5

```
public function setUp()  
{  
    parent::setUp();  
    $this->withoutExceptionHandling();  
}  
  
public function testExceptionHandling()  
{  
    $this->get('/throwsException')->assertStatus(200);  
}
```

Some more available assertions

```
$response->assertSuccessful();
```

```
$response->assertRedirect($uri);
```

```
$response->assertHeader($headerName, $value = null);
```

```
$response->assertCookie($cookieName, $value = null);
```

```
$response->assertPlainCookie($cookieName, $value = null);
```

```
$response->assertSessionHas($key, $value = null);
```

```
$response->assertSessionHasErrors(array $keys);
```

```
$response->assertSessionMissing($key);
```

```
$response->assertSee($value);
```

```
$response->assertSeeText($value);
```

```
$response->assertDontSee($value);
```

```
$response->assertDontSeeText($value);
```

```
$response->assertViewHas($key, $value = null);
```

```
...
```

Let's code!

Further links

- <https://laravel.com/docs/5.4/testing>
- <https://leanpub.com/laravel-testing-decoded>
- <https://laracasts.com/series/phpunit-testing-in-laravel>
- <https://laracasts.com/series/lets-build-a-forum-with-laravel>
- <https://github.com/rkulik/testing-in-laravel>

Thank you!