

Laravel 8 TDD

First create the test class with the following command:

php artisan make:test ArticlesTest

Write the test content and then run it with the command **php artisan test --filter ArticlesTest**.

```
tests > Feature > ArticlesTest.php > PHP Intelephense > ArticlesTest > test_example
1  <?php
2
3  namespace Tests\Feature;
4
5  use Illuminate\Foundation\Testing\RefreshDatabase;
6  use Illuminate\Foundation\Testing\WithFaker;
7  use Tests\TestCase;
8
9  class ArticlesTest extends TestCase
10 {
11     /**
12      * A basic feature test example.
13      *
14      * @return void
15      */
16     public function test_example()
17     {
18         // $response = $this->get('/');
19
20         // $response->assertStatus(200);
21         $data = [
22             'title' => 'Gustavo',
23             'content' => 'Vinicius'
24         ];
25
26         $expected = [
27             'data' => [
28                 'id' => 7,
29                 'title' => 'Gustavo',
30                 'content' => 'Vinicius',
31                 'created_at' => date('Y-m-d\TH:i:s.u\Z'),
32                 'updated_at' => date('Y-m-d\TH:i:s.u\Z')
33             ]
34         ];
35
36         // the dump function shows the code error at console.
37         $this->post(route('articles.create'), $data)
38             ->dump()
39             ->assertStatus(201)
40             ->assertJson($expected);
41     }
42 }
43
```

After the test is written, write the route in the **api.php** file at the folder **routes**.

```
Route::post('articles/create', [ArticlesController::class,
'store'])->name('articles.create');
```

Then run the test again, and it will fail again because the controller is not written yet.

Now create the **Model** and the **Migration** with the command:
php artisan make:model Article -m

```
database > migrations > 2022_08_01_150540_create_articles_table.php > PHP Intelephense >
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class CreateArticlesTable extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create('articles', function (Blueprint $table) {
17             $table->id();
18             $table->string('title');
19             $table->string('content');
20             $table->timestamps();
21         });
22     }
23
24     /**
25      * Reverse the migrations.
26      *
27      * @return void
28      */
29     public function down()
30     {
31         Schema::dropIfExists('articles');
32     }
33 }
```

Add the fillable at the **Article** model file to use the **create** function at the **Controller**.

```
app > Models > Article.php > PHP Intelephense > Article
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Article extends Model
9  {
10     use HasFactory;
11
12     protected $fillable = [
13         'title',
14         'content'
15     ];
16 }
17
```

Now write the controller.

```
app > Http > Controllers > ArticlesController.php > PHP Intelephense > ArticlesController > store
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Article;
6  use Illuminate\Http\Request;
7
8  class ArticlesController extends Controller
9  {
10     public function store(Request $request)
11     {
12         $data = $request->all();
13
14         $article = Article::create($data);
15
16         $result = [
17             'data' => $article
18         ];
19
20         return response()->json($result, 201);
21     }
22 }
```

After building everything to the test pass, run it again, and you will have the following result:

```
[/var/www/html # php artisan test --filter ArticlesTest
{#1618
  +"data": {#1613
    +"title": "Gustavo"
    +"content": "Vinicius"
    +"updated_at": "2022-08-01T15:25:01.000000Z"
    +"created_at": "2022-08-01T15:25:01.000000Z"
    +"id": 7
  }
}

PASS Tests\Feature\ArticlesTest
✓ example

Tests: 1 passed
Time: 1.87s

/var/www/html #
```

More flexible approach:

A more flexible way to build tests is using the `assertJsonFragment`, which verifies if the returned JSON has some specific part in it.

```
$expected = ['title' => 'Gustavo'];
```

```
$this->post(route('articles.create'), $data)
    ->dump()
    ->assertStatus(201)
    ->assertJsonFragment($expected);
```

Image example:

```
tests > Feature > ArticlesTest.php > PHP Intelephense > ArticlesTest > test_example
1  <?php
2
3  namespace Tests\Feature;
4
5  use Illuminate\Foundation\Testing\RefreshDatabase;
6  use Illuminate\Foundation\Testing\WithFaker;
7  use Tests\TestCase;
8
9  class ArticlesTest extends TestCase
10 {
11     /**
12      * A basic feature test example.
13      *
14      * @return void
15      */
16     public function test_example()
17     {
18         // $response = $this->get('/');
19
20         // $response->assertStatus(200);
21         $data = [
22             'title' => 'Gustavo',
23             'content' => 'Vinicius'
24         ];
25
26         $expected = ['title' => 'Gustavo'];
27
28         // the dump function shows the code error at console.
29         // the assertJsonFragment allows you to build a more flexible test.
30
31         $this->post(route('articles.create'), $data)
32             ->dump()
33             ->assertStatus(201)
34             ->assertJsonFragment($expected);
35     }
36 }
```

Test results images:

```
• Tests\Feature\ArticlesTest > example
Unable to find JSON fragment:

[{"title":"Other name"}]

within

[{"data":{"content":"Vinicius","created_at":"2022-08-01T15:43:16.000000Z","id":9,"title":"Gustavo","updated_at":"2022-08-01T15:43:16.000000Z"}}].
Failed asserting that false is true.

at tests/Feature/ArticlesTest.php:32
 28 |         // the dump function shows the code error at console.
 29 |         $this->post(route('articles.create'), $data)
 30 |             ->dump()
 31 |             ->assertStatus(201)
→ 32 |             ->assertJsonFragment($expected);
 33 |         }
 34 |     }
 35 |
```

Tests: 1 failed

Time: 1.61s

```
/var/www/html # php artisan test --filter ArticlesTest
{#1618
  +"data": {#1613
    +"title": "Gustavo"
    +"content": "Vinicius"
    +"updated_at": "2022-08-01T15:42:59.000000Z"
    +"created_at": "2022-08-01T15:42:59.000000Z"
    +"id": 8
  }
}
```

PASS Tests\Feature\ArticlesTest

✓ example

Tests: 1 passed

Time: 1.30s