

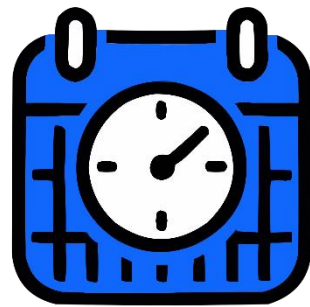
AUDOUARD Raphaël
GUIHENEUF Mattin
MOURGUE Clément
ROUYER Johan
VICTORAS Dylan

TD A – Formation Initiale
TD A – Formation Initiale
TD A – Formation Initiale
TD A – Formation Initiale
TD A – Formation Initiale



S6.01

Valorisation d'une application



DFM

Sujet Don't Forget Me
Commanditaire : M. Antoine MOULIN
Notice de tests

B.U.T Informatique – Semestre 6 (2024 – 2025)

I.U.T de Bayonne et du Pays Basque

Département Informatique

Notice de tests

Nom du projet : Don't Forget Me

Version : 1.0

Dernière mise à jour : 26/01/2025

Cette notice explique la procédure pour concevoir, exécuter et analyser des automatisés avec Laravel basés sur les exemples déjà effectués.

1. Contrôlez l'environnement de tests

Assurez vous que PHPUnit, bibliothèque de tests PHP, est correctement installé via Composer :

```
composer require --dev phpunit/phpunit
```

Vérifiez également dans le fichier phpunit.xml, fichier d'environnement de PHPUnit, que les deux lignes suivantes sont décommentées :

```
<env name="DB_CONNECTION" value="sqlite"/>
<env name="DB_DATABASE" value=":memory:"/>
```

Celles-ci configurent l'utilisation de SQLite en mémoire, garantissant une base de données propre et isolée pour chaque test.

2. Se placer sur la branche adaptée

Placez vous sur la branche correcte du projet avant créer le fichier de tests. Nous avons suivi la nomenclature suivante : test/ID-Ticket. ID-Ticket correspond à l'identifiant JIRA du ticket de la fonctionnalité testée.

Exemple :

```
git switch -c test/DFM-2
```

3. Créer un fichier de tests

Tous les tests Laravel doivent se trouver dans le répertoire tests/. Celui-ci se décompose en deux sous-dossiers :

- Feature : Tests couvrant des fonctionnalités complètes (ici la connexion au compte)
- Unit : Tests ciblant une seule méthode ou logique interne.

Exemple pour créer un fichier de test :

```
php artisan make:test LoginTest
```

4. Configurer le fichier de test

Dans le fichier de test, configurez les imports nécessaires et ajoutez le trait RefreshDatabase pour rafraîchir la base de données avant chaque test. Celle-ci sera différente de celle de l'application puisqu'une version SQLite est utilisée exclusivement pour les tests.

Exemple :

```
namespace Tests\Feature;
use Tests\TestCase;
use Illuminate\Foundation\Testing\RefreshDatabase;
use App\Models\User;
class LoginTest extends TestCase
{
    use RefreshDatabase;
}
```

5. Suivre une bonne démarche d'écriture de test

Chaque fonction de test doit suivre la bonne nomenclature et la démarche GIVEN / WHEN / THEN que nous avons vu. Ne pas hésiter également à utiliser une fonction « Helper » dans le cas où différents types de saisies doivent être testées.

6. Messages d'erreurs

Si besoin de comparer avec les messages d'erreur, utilisez dd() pour voir ceux qui sont ressortis.

Ex :

```
$response = $this->post('/login', ['email' => '', 'password' =>
'']);
dd(session('errors')->all());
```

Sortie possible :

```
array:2 [
    0 => "Le champ adresse e-mail est obligatoire."
```

```
1 => "Le champ mot de passe est obligatoire."  
]
```

Les messages d'erreurs sont centralisés via les fichiers de traduction situés dans le fichier `lang/{locale}/validation.php`. Pour l'instant, nous ne traitons que le français et l'anglais. Ne pas hésiter à rajouter ou modifier ces messages pour qu'ils soient plus clairs si le cas le demander.

Il est aussi possible de changer l'environnement pour tester que les langues fonctionnent avec la fonction `setLocale()`

7. Exemple

Un exemple complet est disponible avec le fichier `LoginTest` pour s'en inspirer.

8. Lancer les tests

Exécutez tous les tests :

```
php artisan test
```

Exécutez un test précis :

```
php artisan test --filter=test_empty_password_should_return_error
```

Avantages des bonnes pratiques

- **Cohérence** : Les messages d'erreur sont centralisés dans les fichiers de traduction.
- **Multilingue** : Les tests fonctionnent quelle que soit la langue configurée.
- **Maintenance facilitée** : Les modifications aux messages d'erreur sont propagées automatiquement.