

# Todolist – Documentation technique

## Environnement du projet

Le langage de programmation utilisé est PHP (version 8.1.3 aujourd'hui)

Le projet a été réalisé avec le framework Symfony (version 6.3 aujourd'hui).

Base de donnée Mysql.

L'ORM utilisé est Doctrine (inclus dans Symfony).

Vous pouvez regarder la documentation officielle de Symfony pour apprendre comment utiliser le framework pour créer des routes, utiliser les assertions, les formulaires, les migrations, créer des templates beaucoup d'autres fonctionnalités.

Pour installer le projet je vous suggère de lire le README.MD.

## Connection - deconnection

Les utilisateurs sont stockés en base de donnée.

L'entité relative aux utilisateurs est "src\Entity\User.php".

Les routes de connection et deconnection "src\Controller>LoginController.php" fichier.

Les deux viennent du framework Symfony.

Si vous voulez mettre à jour les url ou nom de route , vous devrez remplacer "config/packages/security.yaml" dans "firewalls" -> "main" et "form\_login" ou "logout" et également dans le controller.

Si vous avez besoin de plus d'informations, vous pouvez lire la documentation de Symfony: <https://symfony.com/doc/6.3/security.html>

## Test

Le projet utilise le framework PHPUnit.

La documentation Symfony relative a phpunit :

<https://symfony.com/doc/current/testing.html>

Les fichiers de tests sont localisés dans le dossier "tests\" dans le projet.

L'architecture de tous les dossiers et fichiers doit être identique au dossier « src\ ».

Les fichiers de test doivent avoir un suffix "Test" juste avant l'extension.

Par exemple :

Pour tester le fichier “\src\Controller\TaskController.php”,

Vous devrez créer un fichier “\tests\src\Controller\TaskControllerTest.php”.

Aujourd’hui, le code coverage est a plus de 98%.

La configuration de test se fait dans le fichier “phpunit.xml” (à la racine du project).

Les noms des méthodes de test doivent commencer par le mot clé “test” et elle doit être public (exemple : public function testMethodSuccess()).

### Bundles et librairies additionnelles

Utiliser la commande de terminal “composer install” pour installer tous les bundles et librairies.

Ils sont situés dans le dossier “\vendor”.

Les bundles sont listés dans le fichier “\config\bundles.php”.

Les librairies et les bundles sont visibles dans le fichier “\composer.json”.

### Qualité du projet

Sur le repository Github, SymfonyInsight lance une analyse à chaque push.

Aujourd’hui, le rang atteint est Platinum, soyez sûr de garder un bon niveau de rang en résolvant les suggestions lorsqu’il y en a.

Le projet utilise les librairies phpstan & cs fix pour empêcher d’introduire des erreurs et des formats de code non conformes.

Un hook pre-push est configuré et il déclenche une analyse phpstan, une analyse cs fix, et lance tous les tests unitaires pour empêcher toute régression.

Vous pouvez trouver toutes ces commandes pour lancer ces analyses dans le fichier README.MD.

Essayez, du mieux que vous le pouvez, de respecter les bonnes pratiques:

- Don’t repeat yourself (éviter la duplication de code)
- Le code doit être lisible (nommage, complexité ...)
- Suivez les principes SOLID (utiliser les interfaces, services ...)

- Ajouter des tests quand vous créer du nouveau code (considerez l'approche test driven development)
- Gardez la logique coté back (fichier php), et pas dans les templates

Version 1.1 – Date: 13-12-2023