

# P8 TodoList - Description du projet

---

Votre rôle ici est donc d'améliorer la qualité de l'application. La qualité est un concept qui englobe bon nombre de sujets : on parle souvent de qualité de code, mais il y a également la qualité perçue par l'utilisateur de l'application ou encore la qualité perçue par les collaborateurs de l'entreprise, et enfin la qualité que vous percevez lorsqu'il vous faut travailler sur le projet

## Tâches à faire

1. implémentation de nouvelles fonctionnalités
2. corriger les anomalies
3. implémenter les tests automatisés

Correction des anomalies :

- automatiquement, à la sauvegarde de la tâche, l'utilisateur authentifié soit rattaché à la tâche nouvellement créée.
- Lors de la modification de la tâche, l'auteur ne peut pas être modifié.
- les tâches déjà créées, il faut qu'elles soient rattachées à un utilisateur "anonyme".

Tests automatisés :

- les implémenter avec PHPUnit
- prévoir les données de `test` pour prouver le bon fonctionnement dans les cas explicites ici.
- fournir un rapport de couverture de code à la fin du projet. Taux de couverture supérieur à 70%.

Rôle utilisateur

Choisir un rôle pour un utilisateur

- Administrateur (ROLE\_ADMIN)
- utilisateur (ROLE\_USER)

Autorisation

Seuls les utilisateurs ayant le rôle administrateur (ROLE\_ADMIN) doivent

pouvoir accéder aux pages de gestion des utilisateurs.

Les tâches rattachées à l'utilisateur "anonyme" peuvent être supprimées uniquement par les utilisateurs ayant le rôle administrateur (ROLE\_ADMIN).

## Documentation

- Produire une "documentation technique"
- expliquer l'implémentation de l'authentification
- prévoir qu'un débutant puisse : comprendre quel(s) fichier(s) il faut modifier et pourquoi, comment s'opère l'authentification, où sont stockés les utilisateurs.
- Produire documentation des procédures pour tous nouveaux développeurs souhaitant apporter des modifications au projet.
- detail du processus de qualité
- règle à respecter.
- Audit de performance :
- Blackfire obligatoire

## Pré Requis

### Server

besoin de : PHP "php": ">=7.2.5" et phpMyAdmin

Utilisation pour le projet de :

- Apache 2.4.46
- phpMyAdmin 4.9.7
- MySQL 5.7.31
- PHP 7.4.9
- Composer [Link download composer](#)

## Library/ Package

- composer require fzaninotto/faker
- composer require phpunit/phpunit [doc php unit](#)
- composer require dama/doctrine-test-bundle

## Guide d'installation

- Clonez ou téléchargez le repository GitHub dans le dossier :

- composer install

```
git clone(https://github.com/pascalincte91/ToDoList_.git)
```

Configuration du fichier .env

```
APP_ENV=dev
APP_SECRET=3df989c8978c1eb689760998b99d99aa
APP_SECRET=3df989c8978c1eb689760998b99d99aa
DATABASE_URL="mysql://root@127.0.0.1:3306/projet_todoList?serverVersion=5.7"
```

## Créer la base de données et charger les fixtures

```
php bin/console doctrine:database:create
php bin/console doctrine:migrations:migrate
php bin/console doctrine:fixture:load
```

Autres commandes:

Vider cache si besoin: php bin/console cache:clear

## Tests / Php unit

```
Installer et créer la base de données et charger les fixtures

- data base environnements pour tests : (.env.test.local)
DATABASE_URL="mysql://root@127.0.0.1:3306/todoList_test?serverVersion=5.7"

php bin/console doctrine:database:create --env=test
php bin/console doctrine:schema:create --env=test
php bin/console doctrine:fixtures:load --env=test
```

Liste des arguments pouvant être utilisés

```
php ./vendor/bin/phpunit --help
Lancement des tests: php bin/phpunit
exemple : php bin/phpunit tests\TestsFunctionals --testdox
          php bin/console/phpunit --filter [nom de la fonction]
```

## Authentification

Admin: Email : sebastien@gmail.fr Password: pascale

Utilisateur : rene.gros@live.com Password: pascale

Contribution (FR/EN) [Fr](#) / [En](#)