

Comment contribuer au projet ?

Document traduit en anglais, installé à la racine du projet, nommé contribution.md.

1/ Installation du projet ToDo &Co	2
2/ Contribuer au projet	3
a/ Développer de nouvelles fonctionnalités	3
b/ Merger vos modifications	3

1/ Installation du projet ToDo &Co

Ouvrir la console.

Cloner le projet :

- **git clone https://github.com/fleurdeveley/OC_P8_ToDo-Co.git**

Entrer dans le dossier du projet :

- **cd OC_P8_ToDo-Co**

Configurer vos variables d'environnement :

- containers Docker (à la racine du projet) : **cp .env.example .env**
- URL de la base de données : **cp app/html/.env app/html/.env.local && cp app/html/.env app/html/.env.test**

Créer le réseau Docker :

- **docker network create project8**

Lancer les containers :

- **docker-compose up -d**

Entrer dans le container PHP pour lancer les commandes de la base de données :

- **docker exec -ti [nom du container php (container id)] bash**

Installer composer et les dépendances :

- **composer install**

Installer la base de données:

- **php bin/console make:migration**
- **php bin/console doctrine:migrations:migrate**

Installer les fixtures (démon de données factices) :

- **php bin/console doctrine:fixtures:load**

Sortir du conteneur :

- **exit**

Créer une branche pour chaque fonctionnalité avec un nom explicite :

- **git checkout -b *NomDeLaBranche***

2/ Contribuer au projet

a/ Développer de nouvelles fonctionnalités

Afin de proposer un code compréhensif et facilement évolutif, écrire votre code en respectant les standards :

- les classes et les méthodes devront être commentées,
- le code doit avoir une bonne présentation et indentation,
- le code doit respecter les normes PSR (PSR-1, PSR-2, PSR-4 et PSR-12).

Après chaque nouvelle fonctionnalité, réaliser les tests unitaires et fonctionnels en couvrant les méthodes, les fonctions et les fonctionnalités de l'application.

Pour chaque bout de code fonctionnant et testé, commiter sur votre branche :

- en ajoutant : **git add** .
- en commitant : **git commit -m "description rapide de vos fichiers"**
- en poussant votre travail sur votre branche grâce à la commande : **git push origin nomDeLaBranche**

Réaliser une Pull Request :

- sur le repository du projet, cliquer sur New Pull Request,
- comparer la base : merge <- compare:nom de votre branche,
- cliquer sur la flèche à côté de Create Pull Request
- cliquer sur Draft.

NB: le Draft permet de verrouiller la Pull Request tant que votre travail n'est pas terminé.

b/ Merger vos modifications

Vérifier votre code avec :

- le respect des standards (cf a/ Développer de nouvelles fonctionnalités),
- la qualité : soumettre votre code à un contrôleur de qualité grâce à Codacy (ou CodeClimate ou SymfonyInsight),
- la performance : soumettre votre code à un contrôleur de performance grâce à BlackFire,
- faire les corrections si besoin.

Vérifier que tous les tests passent au vert grâce à la commande dans le container php :

- **composer tests**

Cette commande lance les tests unitaires et fonctionnels pour obtenir un rapport du coverage en HTML, se trouvant dans app\html\coverage.

Votre travail sur votre fonctionnalité est terminée :

- enlever le Draft,
- cliquer sur Merge pull request.