

I) Installation du projet

- 1) Ouvrir la console et choisir un dossier pour installer le projet.
- 2) Télécharger le projet avec la commande :
git clone https://github.com/CamileGhastine/ToDoOC8.git
- 3) Vous rendre dans le dossier du projet avec la commande :
cd ToDoOC8
- 4) Installer composer et ses dépendances avec la commande :
composer install
- 5) Ouvrir le fichier **.env** et configurer la connexion à la base de données (ligne 26)
- 6) Créer la base de données et charger les fixtures avec la commande :
Composer prepare

Avant de commencer à coder, créer votre propre branche en la nommant avec un nom explicite, reflétant la fonctionnalité développée :

git checkout -b leNomDeMaBranche

II) Contribution au projet

1) Développement de la nouvelle fonctionnalité

- 1) Ecrire votre code en respectant les standards édictés ci-dessous.
- 2) Ecrire OBLIGATOIREMENT (avant, pendant ou après) les tests unitaires et fonctionnels en essayant de couvrir :
 - tout le code,
 - le maximum de cas limites.
- 3) Effectuer des commits pertinents avec des messages clairs :
git commit -am "mon message"
- 4) Pousser vos commit sur le dépôt distant github :
git push origin leNomDeMaBranche

Standard de code

Afin de faciliter le travail collaboratif, votre code doit suivre quelques règles :

- Les classes et les méthodes doivent être commentées.
Sans en abuser, ne pas hésiter à commenter certaines parties du code si cela vous semble nécessaire.
- Votre code doit être bien présenté et bien indenté.
Au besoin installer PHP-CS-Fixer :
composer global require friendsofphp/php-cs-fixer
Et régler automatiquement les problèmes avec la commande :
php-cs-fixer fix src/monDossier/monFichier.php
- Votre code doit respecter les standards de code de Symfony basés sur les standards [PSR-1](#), [PSR-2](#), [PSR-4](#) et [PSR-12](#) (<https://symfony.com/doc/4.4/contributing/code/standards.html>)
Au besoin, installer PHP-CodeSniffer :

composer global require "squizlabs/php_codesniffer="*

Et corriger pour chaque standard les problèmes soulevés par PHP-CodeSniffer :

phpcbf --standard=PSR12 src/monDossier/monFichier

Important : Afin de ne pas compliquer la relecture du code, ne jamais corriger les standards d'un code que vous n'avez pas écrit.

2) Merger vos modifications

- 1) Si cela vous paraît utile, ne pas hésiter à écrire une documentation sur la fonctionnalité développée.
- 2) Vérifier les standards de code, la qualité et la performance du code (voir ci-dessous).
- 3) Vérifier que tous les tests (ceux existants et les vôtres) passent au vert.
Pour cela, lancer la commande : *bin/phpunit*
- 4) Proposer une pull request :
 - Sur le repository de votre compte Github, cliquer sur "*Compare & pull request*".
 - Compléter le formulaire en choisissant un titre judicieux et en donnant des explications claires sur les changements effectués.
 - Valider la pull request.

Vos propositions de modifications seront étudiées. Si elles sont jugées opportunes, elles seront intégrées au projet.

Qualité du code

Utiliser **CodeClimat** (www.codeclimat.com) pour soumettre votre code à un contrôle de qualité.

Le cas échéant, modifier votre code afin de régler les problèmes soulevés par CodeClimat.

Performance du code

Utiliser Blackfire (<https://www.blackfire.io/>) pour soumettre votre code à un contrôle de performance.

En cas de dégradation de la performance, trouver les solutions d'optimisation afin de régler cela.