Guidelines de collaboration sur le projet TodoAndCo

Sommaire

Introduction

Ce document est à lire avant de commencer à travailler sur le projet ToDoAndCo. Il est à destination des développeurs qui interviendront tout au long de la vie du projet.

Deux aspects majeurs seront développés: la collaboration sur git ainsi que le maintien de la qualité de code selon les standards PSR-1 et PSR-2.

Utilisation du git

Dans une optique de garder une traçabilité dans les différentes évolutions apportées au long de la vie du projet et d'avoir des possibilités de rollback efficaces, nous utiliserons github comme solution d'hébergement et de partage du code.

Récupération du projet

De manière à récupérer le code du projet, il faut dans un premier temps cloner le dépôt. Pour ce faire après avoir installé git sur votre machine il faudra exécuter la commande suivante:

```
git clone https://github.com/Ma77hieu/TodoAndCo2.git
```

Modifications

Pour pouvoir effectuer vos modifications sans impacter le travail des autres développeurs intervenant actuellement sur le projet, la gestion des modifications s'effectue de la manière suivante:

a - Création d'une branche de feature

Pour créer une branche sur laquelle vous pourrez effectuer vos modifications sans modifier le tronc commun (fonctionnel) de l'application, utilisez la commande:

```
git checkout -b my-feature-name
```

Vous venez de créer une nouvelle branche depuis la branche "main" de l'application et vous êtes actuellement situé sur cette nouvelle branche. Toutes vos modifications y seront stockées et elles ne viendront pas impacter le fonctionnement de l'application dans sa version nominale située sur la branche principale nommée "main"

b - Enregistrer les modifications sous forme de commits

A chaque fois que vous avez terminé une tâche listée dans les issues du projet, vous pouvez livrer un commit sur votre branche en indiquant l'issue clôturée par ce commit.

```
git commit -m "closes #issueNbr; your commit message"
```

c - Enregistrer vos modifications dans le dépôt distant

Pour avoir une sauvegarde en ligne de vos modifications et permettre aux autres développeurs d'accéder à votre code si besoin, vous devez "pusher" vos modifications sur le dépôt distant, pour ce faire utilisez la commande:

```
git push origin <your-branch-name>
```

Votre code est désormais disponible en tant que branch sur le dépôt distant.

d - Effectuer une pull request

Une fois que vous avez clôturé l'ensemble des issues qui vous avaient été affectées, vous pouvez soumettre l'ensemble de votre code à la validation des autres développeurs du projet, pour ce faire vous devez aller sur le repo du projet et effectuer une pull request: https://github.com/Ma77hieu/TodoAndCo2/pulls. Vous pouvez vous référer à la documentation officielle de github sur la création de pull request (PR) si besoin.

e - Intégration de vos modifications à la branche main

Une fois le code revu par vos pairs, si celui ci répond aux éxigences du projet en termes de fonctionnel et de qualité de code, votre PR sera acceptée et intégrée directement dans la branche main du projet.

Qualité de code

De manière à s'assurer de la lisibilité du code par un maximum de développeurs, il a été décidé de suivre les standards **PSR-1** et **PSR-2.**

Pour valider le respect de ses standards, il est demandé à chaque développeur intervenant sur le projet de procéder à une vérification automatisée de son code grâce au bundle PHP_CodeSniffer (repo). Pour ce faire il suffit d'installer le projet, d'effectuer les modifications nécessaires et de lancer la commande:

```
./vendor/bin/phpcs --report-full=QualityReport.txt --ignore=/var,/vendor .
```

Quelques précisions sur la commande ci dessus, le flag `--ignore=/var,/vendor` permet d'ignorer les fichiers des bundles externes ainsi que certains fichiers symfony de manière à ne pas voir remonter les erreurs qui n'ont pas été introduites par vos modifications. Il ne vous reste donc plus qu'à corriger les erreurs remontées par le rapport en question avant de push votre branche.