

Projet 13 - Plan de test

Parcours OpenClassrooms - Développeur d'application Python

Étudiant : Guillaume OJARDIAS Mentor : Erwan KERIBIN
Mentor évaluateur : Jimmy Kuassi KUMAKO

Plan de tests

Pour faciliter la mise en place des tests sur le projet, je me suis appuyé sur l'utilisation de deux outils :

- `pytest` pour le framework de tests;
- `coverage` pour calculer la couverture des tests en local;
- `coveralls.io` pour calculer la couverture des tests et l'afficher sur GitHub;
- `tox` pour le paramétrage des environnements de test.

L'application back-end est testé sur plusieurs points :

- les `routeurs`, qui correspondent aux points d'entrées de l'API;
- les fonctions `CRUD`, qui permettent de manipuler les données de l'application;
- les fonctions `utils`;
- les `models`, principalement pour vérifier les relations entre objets.

Chaque nouvelle fonctionnalité fait l'objet de l'écriture d'un test avant sa création (logique de TDD). Pour chaque bug, un test supplémentaire est créé pour couvrir le cas rencontré, puis le code est corrigé.

Les tests sur l'application Python sont lancés avec la commande suivante :

```
tox -e py39
```

Chaque nouvelle fonctionnalité est ajouté au repository distant via une Pull-Request. A la création de la Pull-Request plusieurs GitHub Actions sont lancées :

- tests python;
- tests de la PEP8;
- calcul de la couverture des tests;

La Pull-Request est mergé automatiquement par Mergify lorsque les conditions suivantes sont remplies :

- pas d'erreurs sur les tests Python;
- le code est conforme aux règles de la PEP8;
- la couverture de test est de minimum 85% et elle ne diminue pas de plus de 5%.

NOTE : seule l'application back-end est testée pour le moment.