

Tests Générales et Explications !

Pour faire mes tests, j'ai créé sur chacune de mes applications, un dossier tests qui rassemble l'ensemble des fichiers à tester.

J'ai utilisé le package « **coverage** » pour savoir à combien de % j'étais sur chacun de mes tests et j'ai fini par avoir 97 % de couverture sur l'ensemble de mes tests !

J'ai utilisé le code :

```
$ coverage run --source='.' manage.py test
```

```
$ coverage report
```

Pour mon application « **favorites** » j'ai du tester mes modèles et mes vues. Donc, j'ai créé 2 fichiers : `test_models.py` et `test_views.py`.

Sur le fichier `test_models.py`, j'ai du tester ma classe « **Favorite** », pour ce faire j'ai du créer une catégorie qui avait comme nom « pâte à tartiner », un produit avec tous les composants qu'il lui fallait, un produit de substitue avec une meilleure note de nutrition et un utilisateur. Ce qui m'a permis de réaliser mon test à 90 %.

Sur le fichier `test_views.py`, j'ai du tester 2 fonctions « **favorite_save**, **favorite_list** », j'ai créé une fonction `setUp`, qui regroupe la création d'un utilisateur, d'une catégorie d'aliment et de 2 produits différents de la même catégorie.

J'ai fait un test sur un `status_code` 200 sur ma fonction `favorite_list`.

J'ai fait en sorte de vérifier si le chemin du template était correct et si c'était bien le produit de substitutions qui était enregistré correctement. Ce qui m'a permis de réaliser mon test à 100 %.

Pour mon application « **home** » j'ai du tester mes formulaires et mes vues. Donc, j'ai créé 2 fichiers : `test_forms.py` et `test_views.py`.

Sur le fichier `test_forms.py`, j'ai du tester 3 classes « **ConnexionForm**, **RechercheForm** et **SignupForm** » j'ai donc créé 3 classes de tests différents pour pouvoir les tester, dont 1 avec une création d'un utilisateur et les 3 avec une data et un formulaire pour vérifier si elles étaient valides ou non. Ce qui m'a permis de réaliser mon test à 100 %.

Sur le fichier `test_views.py`, j'ai du tester 2 fonctions « **home**, **legales_notices** », j'ai fait les 2 tests sur un `status_code` 200, pour vérifier que toutes fonctionnent. Ce qui m'a permis de réaliser mon test à 100 %.

Pour mon application « **openfoodfacts** » j'ai du tester mon fichier `utils`. Donc, j'ai créé 1 fichier : `test_utils.py`.

Sur le fichier `test_utils.py`, j'ai du tester ma fonction « **get_json** », j'ai du utiliser le package « **Mock** », j'ai créé 2 produits de la même catégorie avec des `status_code` 200 et 404 pour vérifier que ça fonctionnait correctement. Ce qui m'a permis de réaliser mon test à 100 %.

Pour mon application « **products** » j'ai du tester managers.py, models.py et views.py. Donc, j'ai créer 3 fichiers : test_managers.py, test_models.py et test_views.py.

Sur le fichier test_managers.py, j'ai du tester la classe « **ProductManager** », j'ai créer une fonction setUp, qui regroupe la création de 2 catégories (pâtes à tartiner et pizza), de 3 produits (nutella-ferrero, gonuts-dailyliffe et pizza 4 fromages), pour pouvoir tester que le code fonctionnait correctement. Ce qui m'a permit de réaliser mon test à 100 %.

Sur le fichier test_models.py, j'ai du tester les 2 classes « **Category, Product** », pour ce faire j'ai du créer une catégorie qui avait comme nom (pâte à tartiner) et un produit (nutella-ferrero). J'ai donc ensuite pue tester si le code, le product_name, le nutrition_grade_fr, l'url, l'image_url, l'image_nutrition_url et la category fonctionner normalement. Ce qui m'a permit de réaliser mon test à 88%.

Sur le fichier test_views.py, j'ai du tester 2 fonctions « **research, product_info** », j'ai fais le test sur un status_code 200 pour la fonction (research) et pour (product_info) je lui ai créer une catégorie et un produit pour vérifié qu'il me retournait bien les bonnes informations, je lui ai aussi fait un status_code 200. Ce qui m'a permit de réaliser mon test à 84 %.

Pour mon application « **users** » j'ai du tester mon fichier views.py. Donc, j'ai créer 1 fichier : test_views.py.

Sur le fichier test_views.py, j'ai du tester 4 fonctions « **login, logout, signup, accountlog** », j'ai créer une fonction setUp, qui crée un utilisateur. Puis j'ai fais un status_code 200 avec GET et 302 avec POST, pour la fonction (login). Un status_code 302 pour la fonction (logout). J'ai fais un status_code 200 et 302, pour la fonction (signup) et enfin j'ai fais un status_code 200 pour la fonction (accountlog). Ce qui m'a permit de réaliser mon test à 98 %.