

# Amélioration de projet

---

Les améliorations s'appliquent à l'application web **Pur Beurre** qui recommande aux utilisateurs des aliments alternatifs de meilleur qualité en se basant sur la base de donnée [OpenFoodFacts](#). Cette application est développée avec le framework [Django](#).

## Correctif

### Evolution du projet

Le fichier **views.py** du framework Django contient les fonctions traitant les requêtes HTTP de l'application web. Vue l'accroissement de ce fichier, il devient intéressant de regrouper les fonctionnalités par domaine de responsabilité et ainsi favoriser l'évolutivité du code et sa lisibilité.

### Déplacement de la fonctionnalité de recherche de produits alternatifs

#### Modification

Un dossier **views\_b** est créé afin de regrouper les fichiers contenant les fonctions traitant les requêtes HTTP de l'application web. Dans un premier temps le fichier **research.py** est créé pour regrouper les fonctionnalités liées à la recherche de produits alternatifs.

#### Erreur

En conséquence du déplacement de fonctionnalité, une erreur de redirection survient lors des requêtes de l'utilisateur qui cherche un produit alternatif. Grace aux tests d'intégration existant, l'origine de l'erreur est identifiée.

Le fichier **urls.py** qui lie les requêtes HTTP aux fonctions qui traitent ces requêtes doit ainsi prendre en compte la nouvelle organisation du code.

#### Résolution

Le code est ainsi corrigé et les tests automatisés exécutés sur le serveur [Travis CI](#) passent de nouveau.

```
from .views_b import research as views_research

path('result/<str:user_query>', views_research.result, name='result'),
```

## Nouvelle fonctionnalité

Suite à une requête de l'utilisateur, l'application web effectue les recherches de produits dans sa base de données. Cette dernière ne contenant qu'une fraction de la base de données originelle, il arrive qu'un produit recherché ne soit pas identifié alors que ce produit existe dans la base source d'**OpenFoodFacts**.

### Recherche en ligne

On décide d'ajouter la recherche en ligne d'un produit. Suite à une requête utilisateur, si le produit recherché n'est pas trouvé en base de données locale, il est alors recherché dans la base de données originelle via l'[API OpenFoodFacts](#). Une fois le produit trouvé il est alors ajouté dans la base de données local.

## TDD

Suivant la description de la fonctionnalité de recherche en ligne, un test est ajouté afin de vérifier l'utilisation de l'API de recherche d'OpenFoodFacts et la sauvegarde des informations du produit en base de données. Un mock est utilisé pour simuler l'accès et la réponse de l'API.

Une fois le test mis en place, la fonctionnalité est alors développée puis intégrée au test relatif. Après validation de la nouvelle fonctionnalité par son test, l'ensemble des tests du projet est exécuté sur le serveur d'intégration continu **Travis CI** afin de vérifier la bonne intégration de la fonctionnalité et la non régression du projet.

## Monitoring

Afin de vérifier le fonctionnement en production, les logs sont remontés via [Sentry](#) en fonction des cas d'usage :

- produit trouvé en bas de données locale
- produit trouvé en ligne
- produit non trouvé

## Livrables

[Code Source](#)