

Projet 11 : Evolution de la Plateforme pour amateurs de Nutella

Code source

https://github.com/horlas/OC_Project11

Généralités

Dans le contexte du projet 8 la Plateforme Web fait appel à l'API Openfoodfacts pour la recherche de produits et de substitut de produits. Nous faisons appel à notre base de données seulement pour l'enregistrement des choix utilisateurs.

Contexte

L'entreprise Pur beurre a souhaité nous commander une évolution de la Plateforme Web. Pur beurre souhaiterait avoir une base de données propre à l'application et pouvoir extraire les données OpenFoodfacts. Ces données seraient nettoyées puis enregistrées dans la base. Le chargement de nouvelles catégories pourrait se faire sur simple ligne de commande en notifiant le nom de la catégorie souhaitée. Cet enregistrement pourrait se faire par l'équipe OpenFood Facts.

La recherche utilisateur se fait en priorité sur la base de données Pur_beurre, et si le produit saisi ne s'y trouve pas on bascule la recherche sur Open food Facts.

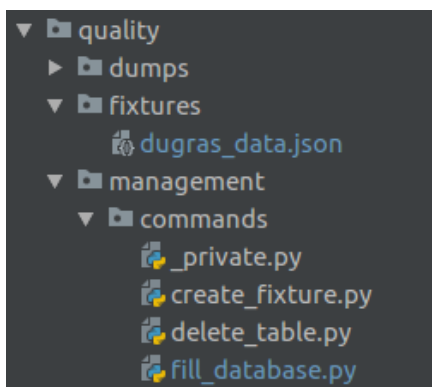
Concernant la recherche d'un substitut, nous divisons la proposition entre Openfoodfacts et Pur beurre : 3 produits de chaque. Ce qui permet d'élargir le choix mais surtout l'enrichissement possible de la base (voir paragraphe suivant)

Nous avons mis en place un bonus ! Lorsque le choix utilisateur de produits sélectionnés et substitués ne se trouvent pas en base propriétaire nous les enregistrons évidemment pour la sauvegarde utilisateur mais aussi dans la table Product de la base. Cet enrichissement fondé sur la recherche utilisateur de la base est transparente pour l'utilisateur.

Nous avons aussi mis en place un programme permettant d'avoir certaines informations sur la base de données pur_beurre.

Choix technologique

Nous avons mis en place des commandes manage.py personnalisées, qui nous permettent de lancer des scripts de création de fixtures et de remplissage de la base en une ligne de commande. Ces scripts se trouvent au niveau de l'app quality dans cet arborescence.



La ligne de commande suivante charge les catégories Pizzas au jambon et Plats préparés surgelés.

\$ python manage.py fill_database 'Pizzas au jambon', 'Plats préparés surgelés'

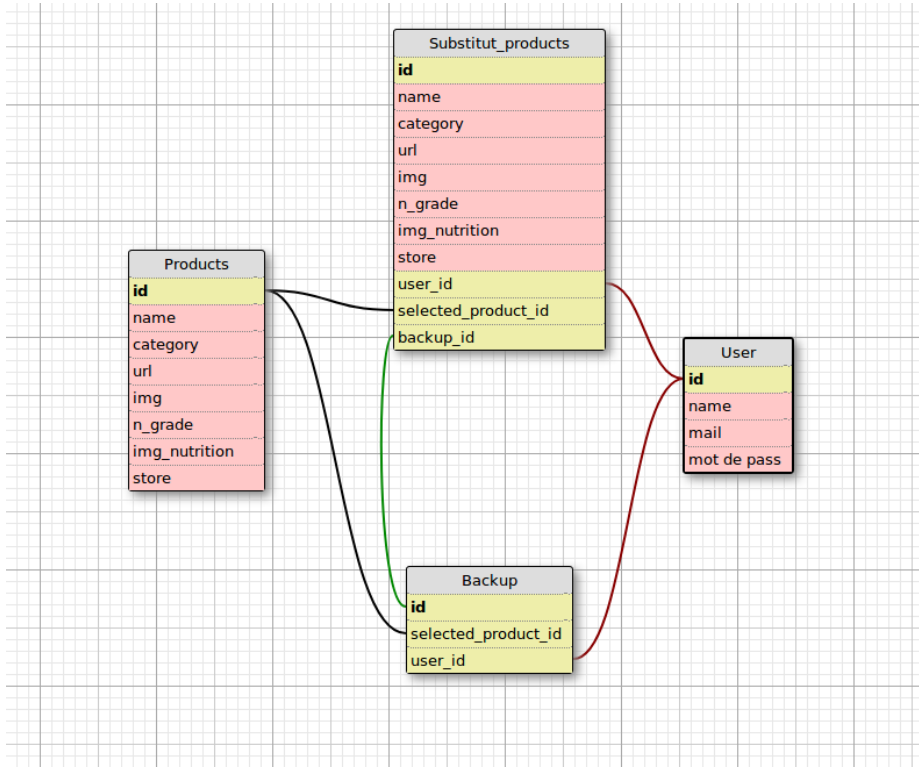
Les entrées en doublons sont supprimées, des retours d'information permettent de savoir combien de produits sont chargés en base, quels sont les produits doublons supprimés, si le processus se déroule correctement

\$ python manage.py database_info

Permet d’obtenir des informations sur la base ex : quel est l’utilisateur qui a fait le plus de sauvegarde. Des données qui peuvent être intéressantes pour faire des statistiques par la suite.

Base de données

Nous avons fait évoluer légèrement la structure de notre base , à présent la table Product comporte tous les champs nécessaires à l’application (y compris , l’image de nutrition et les magasins).



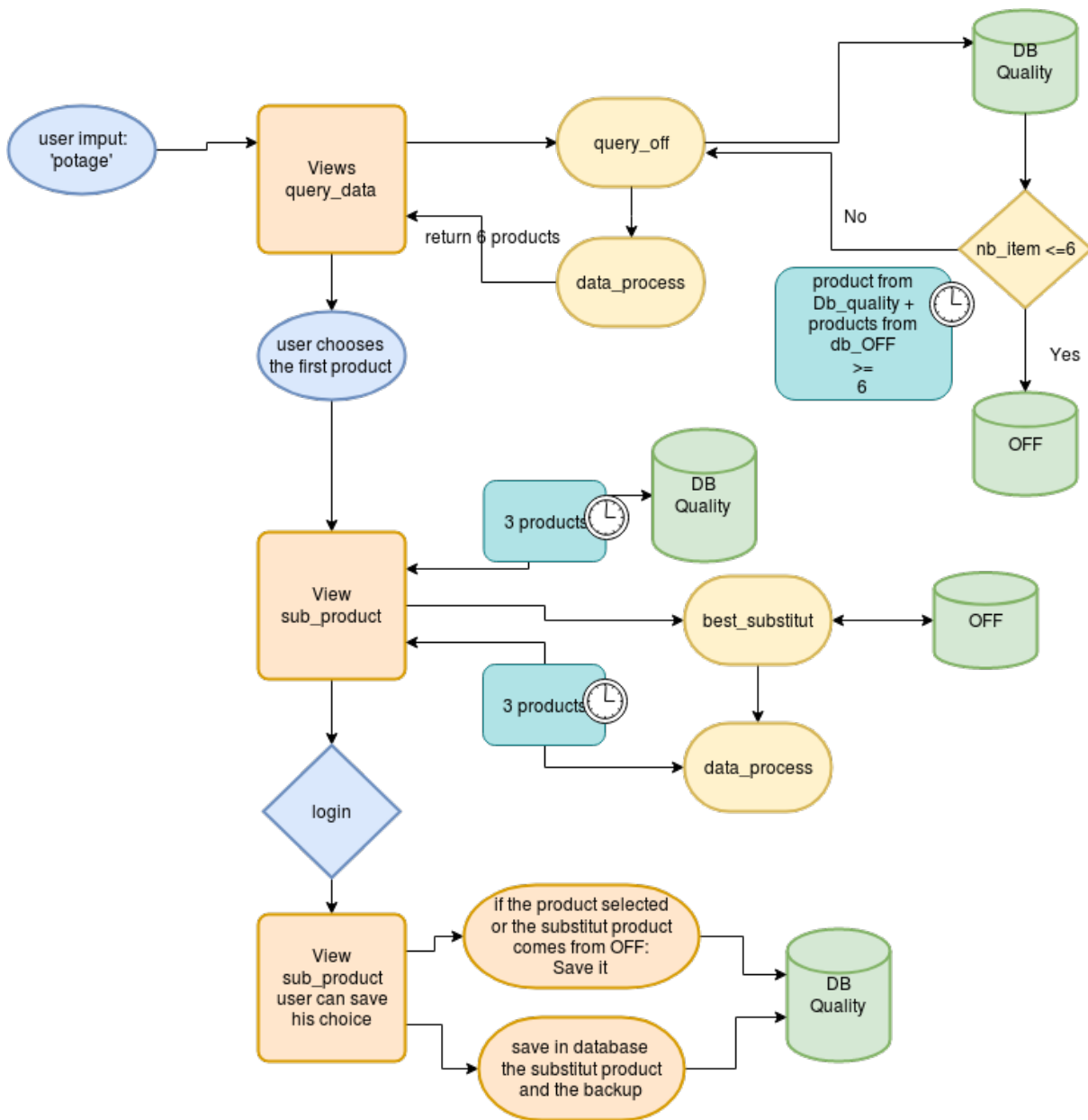
Parcours Utilisateur

Il reste inchangé par rapport au Projet 8.

Algorithme de recherche sur l’API OpenfoodFacts (OFF) :

Nous avons réutilisé l’ensemble de notre code du P8, les fonctions contenues dans le fichier methods.py.

Le schéma suivant décrit l’algorithme.



Affichage des différentes provenances des données

Pour une différenciation de la provenance des données nous transmettons aux gabarits concernés deux types de données dans le contexte, celles ci sont par la suite scannées dans le gabarit.

```

context = {
    'data_quality': data_quality,
    'data_off': data_off
}
return render(request, 'quality/sub_product.html', context)
  
```