

P5 - Utilisez les données publiques de l'OpenFoodFacts

lien github : <https://github.com/jmlm74/P5-OpenFoodFacts.git>

Démarche

Architecture

Le programme est architecturé sous l'architecture MV.C : Modèle – Vue – Contrôleur.

La partie modèle (répertoire models) concerne tout ce qui a trait aux données et à la base de données.

La partie vue (répertoire Views) contient tout ce qui est de l'interface (ici en mode texte). En gros les display.

La partie contrôleur (répertoire controls) contient la logique du programme.

A la racine du projet se trouve le main qui n'est en fait que le démarrage de l'application et un fichier Setup.py qui contient le paramétrage (nom et emplacement de la base de donnée, catégories des produits à récupérer...)

Ce type d'architecture est couramment utilisée dans les développements modernes et par de nombreux frameworks de développement.

Chaque table est déclarée par une classe. Chaque classe fait l'objet d'un fichier du nom de la classe. Les classes présentant les tables ainsi que les méthodes associées (requêtes de recherche par id remontant 1 ou plusieurs lignes) sont stockées dans le répertoire models

La partie menu, tableaux, affichage des détails... est gérée par une classe stockée dans un fichier du répertoire views.

Les interactions utilisateurs et la logique sont stockées dans une classe d'un fichier du répertoire Controllers.

Le module jmlmtools/jmlmtools.py comprend quant à lui des outils tels que clearscreen, la colorisation des menus... ainsi que la connexion à la base de données. Ce module est en fait un module que je réutilise (et enrichit) à chaque nouveau projet.

La base de données

Un fichier PDF des schémas de base est dans le répertoire "doc" du repo github.

Sous MySQL elle est composée de 5 tables dont 2 sont des résultantes d'une relation de plusieurs à plusieurs.

Chaque table s'appelle T_* et dispose d'une clé primaire.

- T_Products → Contient les produits avec leur détail (URL, nutriscore, idcategory). Reliée à la table des catégories par une contrainte sur idcategory.
- T_Category → Contient les catégories (id et libellé) auxquelles appartiennent les produits.
- T_Stores → Contient les magasins (notés brands dans l'API) des produits.
- T_Products_stores → Relation many to many entre T_Products et T_Stores : 1 produit est disponible dans plusieurs magasins et un magasin contient plusieurs produits.
- T_bookmarks → Sert à enregistrer les substitutions. Consiste en fait à une relation de plusieurs à

plusieurs entre les produits : 1 produits peut avoir plusieurs substituts et 1 substitut peut être le substitut de plusieurs produits

La récupération des données se fait grâce à l'API OpenFoodFacts. Les produits sont récupérés par catégorie. Une table temporaire est créé afin de mettre les résultats bruts des requêtes. Ces données sont ensuite légèrement travaillées afin d'éviter les doublons, les données à vide... puis insérées dans les tables définitives. La table temporaire est ensuite supprimée.

Le programme

Le programme a été fait sous forme de package python. C'est à dire qu'il doit être lancé par la commande **python -m** (explications dans le readme.md). Il peut se lancer sans paramètres → lancement du programme principal ou avec les paramètres suivants :

- -d test → teste la connexion, à la base, l'existence des tables et si les données de OpenFoodFacts ne sont pas trop anciennes.
- -d create → (re)crée les tables et les remplit avec les données récupérées via l'API fournie par OpenFoodFacts

Le programme est tout d'abord une suite de menus permettant de choisir une catégorie puis un produit dans cette catégorie.

L'utilisateur peut ensuite voir le détail du produit (ici les nutriscore, l'url et les magasins disposant de ce produit) ainsi qu'un tableau présentant tous les produits de la même catégorie ayant un nutriscore inférieur ou égal. A ce moment l'utilisateur peut choisir un ou plusieurs produits de substitution afin qu'ils soient enregistrés comme favoris.

L'utilisateur peut aussi voir tous les produits qu'il a enregistré en tant que favori. Cela lui affiche un tableau du type produit X → Favori. Il peut alors choisir un produit afin de visualiser le détail du produits et de son ou ses favoris enregistrés.

Les menus ont été colorisés grâce à des "séquences escape" insérées au display. Ils ont aussi été faits avec les modules `prompt_toolkit` et `PyInquirer`
Les tableaux ont été faits grâce au module `prettytable`.

Toutes les urls affichées sont cliquables et renvoient vers le produit sur le site de l'OpenFoodFacts.

Les difficultés

La principale difficulté fut de trouver des données cohérentes sans en récupérer de trop via l'API.

En effet, les données mises à disposition ne sont pas toujours bien renseignées (champs à vides, doublons...) J'ai donc choisi de limiter le nombre de catégories et de produits afin en plus de ne pas surcharger le serveur de requêtes inutiles. Pour cela je me suis limité à 8 catégories et à 3 pages de 20 produits par catégorie en prenant volontairement les pages 2-4 et 6 pour chaque produit.