# Projet n°5 : Utilisez les données publiques de l'OpenFoodFacts

Parcours Développeur d'application -Python

https://trello.com/b/USAysqcC/off

https://github.com/Tepau/OpenFoodFacts

Mon projet a débuté par le suivi des cours concernant les API et les base de données. Ces deux notions étaient totalement nouvelles pour moi.

Après avoir lu la documentation de l'API Openfoodfact, j'ai utilisé Postman pour faire de nombreux tests, qui m'ont permis de comprendre comment envoyer des requêtes via l'API et surtout comment utiliser les données récupérées en fonctions des besoins de mon programme.

J'ai ensuite réalisé un modèle physique de données, afin d'être sûr des données à utiliser et de la façon de les classer dans la Base de données.
Une fois que mon mentor a validé ce modèle physique, j'ai commencé a créer ma base de données, puis à faire des test afin d'y insérer les données récupérées de l'API OpenFoodFact.

Une fois ces notions maîtrisées, j'ai travaillé sur les interactions entre mon programme et l'utilisateur puis , j'ai décidé de découper mon programme en différents fichiers :

- **api_functions.py** : Fichier dans lequel je défini les fonctions nécessaires au fichier service.py. La fonction final_structure permet de convertir les données récupérées dans un format qui permettra une insertion dans la base de données correspondant à mes attentes. Par exemple, concernant le catégories de produits, les produits renvoyés par l'API font obligatoirement parti d'au moins une de mes 5 catégories définies, mais pour autant, l'API me renvoie des catégories que je n'ai pas sélectionné mais auxquelles font quand même parti les produits. Je doit donc créer une fonction me permettant de pour chaque produit de garder uniquement les catégories que j'avais choisi au préalable.

- **service.py** : Fichier qui va récupérer les données utiles à mon programme grâce à l'API. J'ai sélectionné 5 catégories de produits, et j'ai décidé de ne récupérer au sein de ces catégories que des produits ayant comme note nutritionnelle « a » (meilleur note) ou « e » (plus mauvaise note). Une fois les données récupérées je fais appelle à la fonction « final_structure ».

- **functions_db.py** : Je défini toutes les fonctions liées aux requêtes, c'est à dire aux communications avec la base de données. Cela concerne aussi bien les requêtes d'insertion utiles pour le fichier feed_db.py et le fichier main.py (la sauvegarde nécessite une insertion dans la tables « Favorite ») que les requêtes de sélection nécessaires aux fichier main.py lorsque l'utilisateur sélectionne un produit à substituer ou lorsqu'il souhaite accéder à ses recherches sauvegardées.

- **feed_db.py** : Fichier en charge du remplissage de la base de données. Je récupére les données obtenues dans le fichier service.py. J'y appelle les fonctions liées aux requêtes d'insertion définies dans le fichier functions_db.py

- **functions.py** : Je défini une fonctions qui est appelée à chaque fois que l'utilisateur doit faire un choix. Cette fonction utilise un bloc « Try /except » qui permet de ne pas quitter le programme si l'utilisateur fait un choix qui ne correspond pas aux attentes du programme.

C'est à dire que si l'utilisateur peut choisir entre les touches 1, 2 ou 3 et qu'il sélectionne un chiffre différent ou une lettre, le programme lui repose la question jusqu'à ce qu'il donne une réponse valide.

- **main.py**: Je défini une première boucle « main_loop » qui est la boucle principale de mon programme, tant qu'elle tourne le programme est en marche. A l'intérieur de cette boucle, je défini le comportement que doit avoir mon programme en fonction des choix fait par l'utilisateur

- **database.sql** : Contient le script de création des tables en format sql.

- **Constants.py** : Contient les constantes qui vont être utile au bon fonctionnement du programme. J'y défini notamment les catégories que j'ai choisi de représenter dans mon programme, les grades nutritionnels dont j'aurai besoin, ainsi que le questions auxquelles l'utilisateur devra répondre pour avancer lors de l'exécution du programme

J'ai rencontré de nombreuses difficultés lors de la réalisation de ce projet. Notamment au début, j'ai eu beaucoup de mal à comprendre le fonctionnement de l'API, je me suis longtemps perdu dans la documentation et je n'arrivais pas à comprendre quelles étaient les informations qui me seraient utiles.
J'ai aussi eu du mal a rédiger le modèle physique de données, je n'arrivais pas à comprendre la différence avec le diagramme de classe et l'intérêt que de tels diagrammes pouvaient avoir pour la réalisation de mon projet.
Le tableau Trello a également représenté une difficulté, j'avais du mal à me projeter et a prévoir les fonctionnalités nécessaires aux programme. Finalement je l'ai repli au fur et à mesre de mon projet, mais je pense désormais avoir compris son utilité, et son utilisation sera pour moi plus simple lors des prochains projets.
J'ai ensuite « assez rapidement » réussit à rendre mon code fonctionnel, mais tout mon code était dans un seul fichier de 250 lignes, et j'ai eu du mal à « découper » mon fichier en plusieurs en fonction de leur utilité, afin de rendre le code plus lisible.
J'ai rencontré quelques difficultés technique que j'ai réussit à résoudre grâce à de nombreuses visites sur le site stackoverflow.com
Finalement, je suis très content d'être arrivé au bout de ce projet qui m'a pris 2 mois et durant lequel je me suis souvent senti démuni face aux consignes.  Mais j'ai le sentiment d'avoir appris beaucoup de choses que je pourrai mettre en place plus rapidement lors des prochains projets.