

DA Python Projet 5

Données publiques de l'OpenFoodFacts

I) Lien vers le projet 5 sur le compte herodote2242 sur Github :

https://github.com/herodote2242/projet5_openfoodfacts

II) Lien vers tableau Trello :

<https://trello.com/b/9mKdVpHd/p5>

III) Introduction :

Pour des raisons personnelles, je n'ai pas été en mesure de travailler à temps plein sur ce projet, que je termine donc en presque quatre mois, ce qui est beaucoup plus long que ce que j'imaginais. Mais même en avançant un petit peu tous les jours, je n'ai pas abandonné, et finalement le jour de clôture du projet 5 est arrivé.

IV) Rédaction et difficultés rencontrées :

Avant même le début de la rédaction du code du projet 5, un des premiers problèmes auquel j'ai été confronté dans ce projet était un souci de connexion au serveur de MySQL. En effet, en suivant les cours d'OC sur les bases de données relationnelles, en souhaitant installer MySQL, je me suis retrouvé dans l'incapacité d'interagir avec le serveur lors des requêtes. Après plusieurs jours d'essais infructueux, de désinstallations et de réinstallations, mon mentor m'a conseillé d'installer MAMP. Excellente solution, car je n'ai plus eu aucun souci de connexion par la suite.

La première étape de ce projet a été d'imaginer le script de création de la base de données en identifiant les tables nécessaires. Le script de création (creadb.py) dispose d'une fonction de suppression si une base de données ayant le même nom existe déjà, et d'une autre fonction de nettoyage de tables pour les cas où le module est lancé plusieurs fois successivement dans le processus de rédaction de code. L'utilisateur final n'a besoin de lancer ce module qu'une seule fois. Il en va de même pour le module de remplissage de la base de données (feeddb.py), il contient une fonction de nettoyage des tables s'il est lancé plusieurs fois.

Une des premières difficultés rencontrées a été de trouver comment construire la requête de manière efficace pour le remplissage de la base de données grâce à l'API, opération sur laquelle je suis resté bloqué un certain moment. Pour m'aider dans cette tâche, je me suis servi de Postman afin de vérifier comment étaient agencées les données de l'API.

La deuxième étape a été d'identifier de la manière la plus simple et intuitive qui soit les différents menus disponibles à la navigation dans l'application. Outre les informations du briefing du projet, il ressortait donc comme menus : un menu d'accueil principal, un menu de sélection de catégories, un menu de sélection de produits à remplacer, un menu de sélection de produits remplaçants et un menu de choix pour le produit remplaceur d'un côté ; et de l'autre côté un menu de sélection de favoris, un menu de choix pour le favori. Le but étant de pouvoir naviguer dans un menu de plus en plus détaillé, mais aussi de pouvoir revenir au menu précédant, au menu d'accueil ou même de quitter l'application quand l'utilisateur le souhaite, j'ai donc intégré ces différentes possibilités dans chacun des menus.

La troisième étape importante a été de créer le fonctionnement de l'application en elle-même. J'ai commencé par faire un script relativement long, à base de structures if, when, for, imbriquées les unes dans les autres pour correspondre à la navigation entre les menus. Bien qu'opérationnel, le code était très lourd et s'étalait sur près de 150 lignes en continu. A la demande de mon mentor, il a été complètement refondu pour correspondre à une programmation orientée objet. Il est maintenant organisé sur de plus nombreux modules, mais plus courts, dont chacun est responsable d'une seule action.

Enfin, lors des tests complets de l'application, c'est à dire en faisant tourner les deux premiers modules de création de la base de données et de son remplissage, j'ai été étonné de trouver des catégories qui ne contenaient que très peu voire aucun produit, selon les catégories et selon les notes nutritionnelles. Pour éviter qu'un utilisateur fasse une recherche sur une catégorie et ne voit aucun résultat, et/ou pour éviter qu'un utilisateur fasse une recherche sur des produits de substitution et ne voit aucun résultat, une solution a été trouvée. Il s'agit d'ajouter une boucle lors de la recherche de produits dans la base de données OpenFoodFacts avec en critère la note nutritionnelle. Ainsi, au lieu de chercher mille produits de chaque catégorie, l'application en recherche cinq-cents par note et par catégorie.

Une fois tous ces points complétés et résolus, l'application est testée une dernière fois pour vérifier que toutes les fonctionnalités marchent correctement. En plus des modules rédigés en python, le projet contient ce document, le lien vers le tableau Trello qui m'a accompagné tout au long du projet, le lien vers le code stocké sur Github, et un modèle physique de données représentant l'architecture de la base de données.

En route pour le projet 6.