

OPENCLASSROOMS PROJET 70C

Créer GrandPy Bot, le papy robot

Jimi Bourgeois

Github : https://github.com/Bibjim/PAPY_BOT_P7

Trello : <https://trello.com/b/QyvrCfGg/ocp7papybot>

Website : https://github.com/Bibjim/P5_OpenFoodFacts_OC

- But du projet:

Développer une application web qui donne une âme à notre Papy Robot !

Récupérer les coordonnées GPS de l'API Google Maps et les informations de l'API Wikipédia correspondant à la recherche utilisateur et affiche la map et l'un des articles Wikipédia en retour.

- Le projet a été réalisé en plusieurs étapes:

- L'analyse des données des API Google maps et de Wikipédia
- Initialisation de Flask et de la requête Ajax
- La création de la structure du programme
- L'écriture des scripts du programme python et javascript
- Mise en forme de la page Html avec CSS et Bootstrap
- Mise en production sur Heroku

- Le but étant de renvoyer les informations de Google maps et de Wikipédia, j'ai commencé par explorer la construction des fichiers .json générés par les API.
- Afin de retourner les bonnes data à la réponse Ajax, j'ai commencé par créer un script python qui renvoie les données de latitude et de longitude afin de les exploiter à la fois pour la map et pour effectuer également une recherche sur Wikipédia.
- Avec ces coordonnées GSP j'ai pu effectuer dans un premier temps une recherche de la page Wikipédia correspondante aux coordonnées pour ensuite sélectionner l'article à retourner dans la requête ajax.

- Le projet impose de retourner les informations dans une requête Ajax afin d'éviter de recharger la page html. La difficulté était de de garder sur la page les recherches précédentes renvoyées par la requête. Mon mentor m'a donc recommandé d'utiliser la fonction 'last' en passant par une variable pour que les recherches effectuées reste sur la page html comme sur une discussion chat.
- L'autre difficulté était sur les tests. Les tests me retournaient une erreur 'Out of range in index'. Je ne comprenais pas pourquoi et pour cela mon mentor ma demander de modifier l'organisation de mon code pour qu'il soit plus clair et en utilisant le 'Framework geocoder' à la place du 'Framework requests' pour la recherche des coordonnées GPS. Après cette modification, les tests n'étaient toujours pas bon, je ne savais toujours pas pourquoi 'pytest' me retournait cette erreur. J'ai donc utilisé ma key API Google en dur dans le code pour effectuer mes tests et l'erreur venait bien de la clé API. Je l'ai donc mis dans mes variables d'environnement pour réaliser les tests sans avoir la clé en dur dans le code.
- J'ai rencontré le même problème à la mise en production sur Heroku, il suffisait simplement d'ajouter la clé dans l'environnement Heroku (>>>Heroku config:set 'keyname=key').