

Projet 7 Openclassrooms Parcours développeur d'application Python

Créez GrandPy Bot, le « Papy-robot »

<https://grandpybot18.herokuapp.com/>

<https://github.com/Tepau/GrandPyBot>

Ce projet abordait de nombreuses notions totalement nouvelles pour moi, j'ai donc commencé par suivre les cours sur HTML, CSS, JavaScript et JQuery. Cette première phase d'apprentissage a été longue mais très enrichissante.

Une fois ces notions maîtrisées, j'ai commencé à coder sans utiliser Flask. Je me suis donc concentré sur l'interface utilisateur de mon site en utilisant HTML et CSS. J'ai également appris à prendre en main l'API GoogleMaps qui me permet de récupérer l'adresse et les données de géolocalisation d'un lieu, et d'afficher une carte avec un marqueur; et l'API Wikipédia qui me permet de récupérer le contenu d'une page Wikipédia.

Pour moi, la compréhension de l'API Wikipédia a été bien plus simple que l'API GoogleMaps, pour laquelle j'ai dû suivre de nombreux tutoriels avant de bien la comprendre. J'ai hésité à plutôt utiliser « OpenStreetMap » qui est l'une des alternatives les plus connues à GoogleMaps.

Je me suis ensuite concentré sur la mise en place d'un Parser. J'ai décidé de mettre ma liste de « Stop-words », que j'ai récupérée à l'adresse indiquée par Openclassrooms, dans un fichier « .txt » pour avoir un aperçu plus simple du fichier. J'y ai ajouté quelques mots en rapport avec le projet comme « Grandpy » ou « adresse », ainsi que des caractères de ponctuation.

Mon raisonnement pour la mise en place du Parser a été le suivant : Il faut d'abord parcourir la phrase saisie par l'utilisateur et remplacer les éléments qui ne sont pas des lettres par un espace. Ensuite je converti cette phrase en liste de mots, puis je ne garde que les mots qui ne sont pas dans mon fichier « stopwords.txt ». J'ai réalisé de nombreux tests et je trouvais toujours un moyen de faire bugger mon parser ou tout du moins qu'il ne me renvoi pas les mots attendus, notamment lors de la saisie de mots dans une langue étrangère ou avec une faute d'orthographe. Sur les conseils de mon mentor, je suis parti du principe que l'utilisateur écrirait dans un français sans faute.

A ce moment là j'avais donc mon interface utilisateur qui était prête, je maîtrisais les API GoogleMaps et Wikipedia et j'arrivais à parser la question de l'utilisateur pour en garder que les mots utiles pour mes recherches. Seulement, toutes ces fonctionnalités étaient indépendantes et ne communiquaient pas entre elles.

J'ai donc suivi un cours pour prendre en main « Flask ». Au début j'ai trouvé ça déconcertant, notamment la syntaxe légèrement différente en ce qui concerne les fichiers « .html » avec l'ajout des sections de code Jinja2 . J'avais mis pas mal de temps à maîtriser ces notions et j'étais très content d'y parvenir et j'ai eu l'impression au commencement que beaucoup de ces notions étaient remises en cause avec l'utilisation de Flask. Le principe des dossiers « templates » qui contient et « static » qui contient les fichiers « .css » et « .js » a été plutôt simple à comprendre. La notion de « route » également avec l'utilisation de la méthode « render_template » qui va permettre de faire le lien entre le .py et la page HTML prédéfinie.

Une des parties les plus compliquée dans ce projet a été de comprendre comment traiter une requête AJAX avec mon code python. J'ai mis pas mal de temps avant de comprendre qu'il fallait créer une nouvelle route qui allait recevoir les données renvoyées par la requête POST lors de la

validation du formulaire pour l'utilisateur. Finalement, grâce à mon mentor je pense avoir compris le principe suivant : la requête est envoyée sur une « adresse », c'est à cette adresse que mon code python va traiter les données récupérées puis renvoyer les données finales sous format « json » pour pouvoir les retravailler en javascript et afficher les informations souhaitées à l'utilisateur.

Concrètement, dans mon projet, la question saisie par l'utilisateur est transmise à la page « /api/map/ », mon programme va parser la question pour en garder que les mots qui nous importent. Ces mots sont utilisés pour récupérer l'adresse et la position géographique d'un lieu grâce au géocodage de l'API GoogleMap, ainsi que pour récupérer les informations Wikipédia concernant une adresse postale. Ces informations sont renvoyées au format json et sont retravaillées en javascript pour afficher à l'utilisateur, l'adresse du lieu qu'il recherche, une carte Google avec un marqueur correspondant à l'adresse et les 3 premières phrases de la page Wikipédia correspondant à l'adresse du lieu.

Une fois les données récupérées j'utilise JavaScript pour accéder à des éléments du DOM afin de gérer l'affichage de toutes les informations utiles.

Je n'ai que très peu utilisé jQuery car j'ai mis du temps à intégrer le principe du parcours du DOM avec JavaScript et une fois assimilé cela me paraissait plus instinctif de procéder de cette manière.

La mise en place des tests utilisant des mocks pour les API n'a pas été une chose aisée. Je ne comprenais pas l'intérêt de faire ces tests car j'avais l'impression que pour chacun de mes modules je réalisais déjà des tests à l'aide des lignes de code suivant « if __name__ == '__main__': », les explications de mon mentor m'ont été sur ce point essentielles.

Enfin, la mise en ligne à l'aide d'« Heroku » a présenté une difficulté majeure concernant mes API keys. Je les avais d'abord écrites dans un fichier « constantes.py » que je plaçais dans un .gitignore afin que ces données ne soient pas accessibles à tous sur github, seulement, en procédant de cette manière heroku n'y avait pas accès non plus. J'ai donc dû placer les valeurs de mes API keys dans des variables d'environnement de heroku afin d'y avoir accès en toute sécurité sans avoir à les publier sur github.

Pour conclure, j'ai trouvé ce projet très enrichissant, j'ai l'impression d'avoir appris énormément de nouvelles notions.

J'ai beaucoup aimé l'apprentissage de HTML, CSS et JS car j'ai pu faire de nombreux tests concrets qui m'ont permis d'avancer pas à pas avec de plus en plus de certitudes.

J'ai également l'impression de mieux maîtriser des notions que j'aborde depuis le début de la formation, notamment l'utilisation des classes qui jusqu'alors restait floue.

J'ai passé 2 mois à réaliser ce projet, et je suis fier d'être parvenu à ce résultat. Je pense que je dois encore beaucoup m'améliorer dans la compréhension des différentes documentations car j'ai parfois l'impression de perdre un temps fou pour trouver des informations.