

Projet 7 : Créez GrandPy Bot, le papy-robot

Vous pouvez trouver ma "To do List" sur trello.com, le lien [github](#) et le [website](#).

❖ Organisation du travail :

Après avoir lu les cours disponibles sur openclassrooms et pris connaissance du framework Flask, j'ai pu organiser mon travail et le découper en 3 user stories (voir tableau Trello pour plus de détails):

- Créer un environnement Flask et renvoyer une page html simple mais répondant aux fonctionnalités demandées dans le cahier des charges (carte google map, parsing de la question...).
- Décorer le site et ajouter des fonctionnalités en javascript (création du dialogue entre GrandPy et l'utilisateur...).
- Déployer le site sur heroku et corriger les bugs (vérifier que le site fonctionne sur n'importe quel système d'exploitation et sur mobile).

❖ Réalisation :

Lors de la conception de l'application, j'ai appliqué le Test Driven Development et l'ajout pas à pas de nouvelles fonctionnalités.

Exemple de test du parser :

on attend le mot « openclassrooms » après filtrage de la question: "Salut GrandPy ! Est-ce que tu connais l'adresse d'OpenClassrooms ?" ensuite je commence la construction du parser (parser.py) puis vérifie si le test fonctionne et intègre la fonctionnalité à l'application.

J'ai préféré fonctionner par ordre de priorité : faire en sorte que la partie back-end soit opérationnelle avant de peaufiner le front-end du site.

Au niveau de la réalisation, j'ai choisi de développer dans un environnement virtuel avec les dossiers classés ainsi :

- « managers » : contient les classes responsables du parser et de l'appel des API de google maps et wikipedia.
- « templates » : contient les documents HTML.
- « static » : contient toute la partie front-end du site (photos, feuilles de style, javascript).
- « test » : tous les tests utilisés pour vérifier l'application.

❖ Problèmes rencontrés :

Pour réaliser ce projet j'ai rencontré deux principaux problèmes :

- L'approche Test Driven Development : il était difficile parfois de prévoir un test dont on ne savait pas à l'avance quelles librairies (ou méthodes) j'allais utiliser pour réaliser la fonctionnalité voulue. De temps en temps j'ai dû modifier mes tests après rédaction de la fonctionnalité.
- Javascript : Étant débutant dans la création de sites web, j'ai passé un certain temps dans l'apprentissage de ce nouveau langage et son intégration au site web.