

PARCOURS OPENCLASSROOMS DÉVELOPPEUR D'APPLICATIONS PYTHON

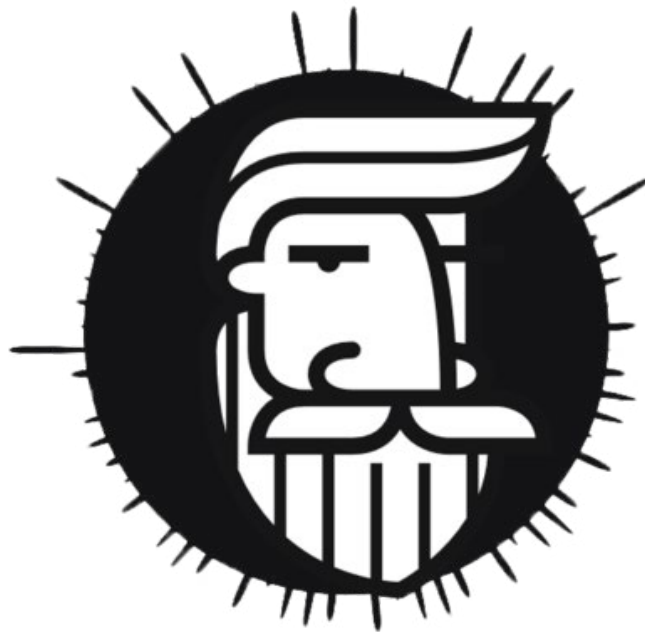
QUENTIN LATHIÈRE

Projet 7 : GrandPy Bot

https://github.com/Synkied/OC_Projet-7-vue

Lien vers l'app : <http://grandpybot-vue-ql.herokuapp.com/>

Lien vers le kanban : <https://tree.taiga.io/project/synkied-granpy-bot/kanban>



Le présent document résume la réalisation du 7^e projet (GrandPy Bot) dans le cadre de mon parcours Développeur d'application - Python.



Table des matières

Définition du projet.....	3
Description.....	3
Fonctionnalités.....	3
Contraintes.....	3
Pré-projet.....	3
Environnement de développement	3
Anticipation des problèmes.....	3
Première difficulté rencontrée.....	3
Solution apportée.....	3
Mise en place du projet.....	4
Démarche choisie	4
Pourquoi l'utilisation de VueJs plutôt que jQuery ?.....	4
Difficulté rencontrée	4
Solution apportée.....	4
Difficulté rencontrée	5
Solution apportée.....	5
Post projet.....	5
Compétences acquises.....	5
Compétences renforcées	5



Définition du projet

Description

Ce projet consiste à créer une application permettant à ses utilisateurs de chercher une adresse, en obtenir la localisation sur une carte Google Maps et des infos concernant ses alentours via Wikipédia.

Fonctionnalités

- Utiliser Python 3 ;
- Requêtes Ajax ;
- Utiliser les API Google Maps et Wikipédia.

Contraintes

- Versionner son code en utilisant Git et le publier sur Github ;
- Développer dans un environnement virtuel en utilisant Python 3 ;
- Respecter les bonnes pratiques de la PEP 8 ;
- Code écrit en anglais : nom des variables, commentaires, fonctions...

Pré-projet

Environnement de développement

- Machine sous Windows 10. Environnement virtuel via virtualenv ;
- Sublime Text 3 avec plusieurs plugins (Principaux : Anaconda, GitGutter) ;

Anticipation des problèmes

- Préparer le projet en TDD (Test Driven Dev) ;
- Structurer les classes et leurs interactions ;
- Apprendre à utiliser Flask ;
- Choisir et utiliser un framework JavaScript ;
- Apprendre à faire des requêtes Ajax ;
- Déployer sur Heroku.

Première difficulté rencontrée

- Créer des tests avant d'avoir créé le code.

Solution apportée

- Suivre le cours OpenClassrooms et différents autres cours, afin de bien intégrer cette notion de TDD, et en comprendre les tenants et aboutissants.



Mise en place du projet

Démarche choisie

Comme demandé, j'ai d'abord écrit les démarches que je pensais effectuer pour ce projet, et l'ordre dans lesquels je pensais les réaliser. Pour me rappeler des tâches à effectuer, j'ai créé un Taiga et me suis fait des User Stories.

J'ai ensuite suivi la démarche TDD et écrit mes tests avant d'écrire les fonctions correspondantes, en vue de faire les étapes : **red**, **green**, refactor.

Dans un même temps, je me suis formé à l'utilisation de *Flask* en suivant le cours OpenClassrooms et divers autres cours (sur YouTube notamment).

Une fois la partie backend (*Flask/Python*) aboutie, je me suis concentré sur la partie Frontend, que j'avais pour le moment délaissée (j'avais simplement fait des tests manuels de fonctionnalités et inclus quelques styles via *Bootstrap*).

J'ai donc dû me former sur *VueJS*, via notamment un très bon cours sur Udemy (*Vue JS 2 - The Complete Guide*) afin de pouvoir utiliser ce framework qui m'était inconnu une semaine avant. Bien heureusement, j'avais déjà fait du *Javascript* et *jQuery* auparavant et possédait donc déjà de bonnes bases sur lesquelles m'appuyer pour apprendre rapidement (et le cours sur Udemy est de vraiment bonne qualité).

Après avoir compris le système de gestion de composants de *VueJS*, j'ai commencé à refactorer mon code et faire une deuxième version de ce projet, séparant réellement le frontend et le backend, et surtout tirant parti de *Webpack* pour construire le frontend.

Pourquoi l'utilisation de VueJs plutôt que jQuery ?

Je voulais tout simplement me lancer un défi, et surtout apprendre un nouveau framework JavaScript, très réputé en ce moment et de plus en plus utilisé.

J'aime par ailleurs l'approche de *VueJS*, qui est soutenu par une communauté de passionnés et non pas par une entreprise des GAFA, même si cela est aussi sa fragilité.

Difficulté rencontrée

- Choisir et utiliser un framework JavaScript. Je souhaitais utiliser un des nouveaux frameworks (React, VueJs, Angular)

Solution apportée

- Se renseigner sur les différences de chacun de ces framework et lire des comparatifs. Discuter avec mon mentor m'a aussi permis de choisir le framework que j'utiliserai : *VueJS*.



Difficulté rencontrée

- Comprendre comment utiliser et surtout configurer Webpack pour faire un paquet regroupant la partie frontend et le lier au backend.

Solution apportée

- Chercher sans cesse sur Google et essayer plusieurs solutions, pour finalement demander à mon mentor qui m'a orienté vers la solution, et m'a permis de chercher avec les bons mots clés. J'ai ensuite fini par comprendre la logique de Webpack et, surtout, l'ai configuré au mieux pour mon projet.

Post projet

Compétences acquises

- Utiliser un nouveau framework JS : VueJs ;
- Utiliser Flask ;
- Déployer sur Heroku ;
- Utiliser Webpack ;
- Structurer un backend et frontend.

Compétences renforcées

- Utiliser des API ;
- Utiliser plusieurs librairies tierces et lire des documentations ;
- Structurer un projet ;
- Écrire un Readme ;
- Écrire un makefile ;
- Écrire de la doc.