

Serguei Nikolaev

Projet 7

Créer GrandPy Bot, le papy-robot

Accès au dépôt projet Github : https://github.com/SergueiNK/oc_p7_papybot.git

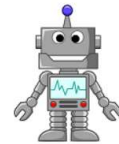
Accès à l'application web : <https://papybot-spatial.herokuapp.com/>

1. Description du projet.

L'idée de ce projet est créée une page web qui permettrait à l'utilisateur de faire une recherche de lieux. Cette recherche de lieux doit donner comme résultat une mape, une adresse et un descriptif issue de wikipedia. Tout cela implémenté d'une petite dose d'humour, car c'est papybot qui donne les informations.

2. Etapes de réalisation du projet.

- * La réalisation de ce projet a été faite en plusieurs étapes.
- * Création de l'environnement de travail avec :
 - Génération d'un dépôt local et distant Github du projet
 - Mise en place de l'environnement virtuel
- * Etude du besoin d'utilisateur :
 - Etude du cahier de charge du projet
 - Etude de la documentation des APIs
 - Réalisation des Users stories
- * Ecriture d'algorithme du logiciel
- * Réalisation de la partie de la fronte de la page d'accueil avec le formulaire utilisateur grâce à HTML , CSS .
- * Réalisation de la partie back :
 - Utilisation de la méthodologie TDD
 - Réalisation de la partie serveur Ajax
 - Requêtes et traitement d'API



Papybot

Serguei Nikolaev

* Réalisation de la partie finale du front :

- Réception et l’affichage des données traités JQuery
- Utilisation de Bootstrap pour la page de retour des informations.

3. Structure du programme.

Le programme peut être distingué en deux parties. Partie front est responsable de l’affichage du site. Tandis que la partie back s’en occupe de la partie serveur avec la gestion de requêtes d’API et de traitement des données.

Dans la partie front, le module HTML est en charge de la structure du site. Son design est assuré grâce aux CSS et du module Bootstrap. La partie transmission des données pour l’affichage est réalisée via module Js incluant des fonctions JQuery.

La partie back est réalisé en Python avec la partie de gestion de requêtes serveur via Flask. La mise en place du controller assure la fluide et la simplicité dans la transmission des commandes et des données.

4. Les difficultés rencontrées

Ce projet m’a permis de s’initier à la partie web avec l’apprentissage de multiples nouveaux langages et de méthodes. Toutefois, il n’était pas évident d’organiser et synthétiser ces connaissances pour les appliquer au projet.

TDD est une méthode puissante, mais j’avais eu du mal à comprendre l’application et le fonctionnement des mocks.

5. Améliorations futures

1. Amélioration du parser qui a tendance à fournir l’ordre des mots du résultat de façon aléatoire.
2. Renvoyer le status erreur et son descriptif de l’API depuis le back vers JS de façon non transcrite par Python. En cas de maintenance, les développeurs pourront avoir accès au type d’erreur depuis la console du navigateur.