Grand-Py Bot

Après avoir lu tous les cours, j'ai commencé ce projet avec une idée qui était présente dans le cours de Boostrap, utiliser un Boilerplate afin de ne pas partir de 0, et pour gagner du temps.

Après avoir vérifié que les technologies utilisées par cet outil était à jour (comme Bootstrap) et agencé correctement le projet selon le cours de Flask, je me suis attaqué à la partie front-end.

Pour ce faire j'ai repris le cours de Bootstrap afin d'avoir une page d'accueil (contenant la barre de recherche) « responsive ».

J'ai par la suite réfléchi à une manière de récupérer des informations de la part de Wikipédia, via l'API de Wikimédia. Le petit problème rencontré était que je n'avais aucune idée de comment récupérer l'introduction d'une page seulement. Afin de résoudre ce problème et d'alléger mon code côté back-end, j'ai utilisé un « wrapper » Python pour cette API qui, non seulement résous le problème, mais permet aussi quelques fonctionnalités telles que l'affichage aléatoire d'une page dans la situation où l'utilisateur ne rechercherais rien de particulier.

L'étape suivante a donc été la présentation du résultat, encore une fois grâce à Bootstrap, d'une partie le résumé de la page Wikipédia, de l'autre la carte avec Google Map. L'utilisation de Google Map aura donc été relativement simple puisque l'API ne prendra simplement que le nom de la page Wikipédia comme argument.

En revanche, ce procédé bien que simple et efficace pose un problème, la contrainte de l'utilisation des Mocks dans le projet, car via ce système nous ne recevons aucune donnée, la carte est totalement indépendante, donc aucune donnée à imiter.

J'ai finalisé le projet par la création des tests unitaire afin de tester simplement les différentes page du site en fonction de différente requête.