

## Grand-Py Bot

Après avoir lu tous les cours, j'ai commencé ce projet avec une idée qui était présente dans le cours de Bootstrap, 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 étaient 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 ensuite créé les tests pour tester m'a future classe Parser qui va s'occuper de garder l'essentiel de la recherche utilisateur et associé une coordonnée à cette recherche via utilisé l'API Google Map GeoSearch.

Après avoir mis en place la classe Parser et vérifié que tout le test était positif, j'ai créé les tests concernant ma classe WikiSearch qui va se charger d'utiliser les coordonnées pour trouver un sujet sur Wikipédia associé au lieu via l'API, puis ensuite utiliser le résultat de cette recherche pour obtenir un bref résumé de cette page.

Une fois le système de recherche aboutie, je me suis attaqué à la relation Client/Serveur, j'ai donc créé la partie « API Rest » du serveur pour renvoyer un JSON à partir d'une recherche fait par le client.

Enfin, j'ai créé le JavaScript permettant d'envoyer la requête concernant la recherche du client au serveur via AJAX ainsi que l'affichage du résultat.

Après avoir vérifié que les tests vis-à-vis des requêtes serveur soit positif, j'ai peaufiné un peu la mise en page du site et la propreté du code.

Lien du site : <https://grandpy-bot.herokuapp.com/>

Lien du Github : [https://github.com/Hideky/Grand-Py\\_Bot](https://github.com/Hideky/Grand-Py_Bot)

Lien du Trello : <https://trello.com/b/BxIA8SjS/projet-7-grand-py-bot>