

Projet 7 - RickPyBot

Mise en place du projet

Avant toutes choses, il faut commencer par installer une base saine pour pouvoir commencer à travailler. J'ai commencé par ajouter les tâches que je devais effectuer dans un tableau Trello (<https://trello.com/b/UuM89UYG/rickpybot-projet-7-ocr>) . Après avoir effectué cela, j'ai créé un *repository* sur Github (<https://github.com/subteno-it-mga/Projet7-GrandPyBot>).

Initialisation de Flask

Comme vu dans les cours qui précèdent et sont nécessaires au projet, j'installe Flask ainsi qu'un environnement virtuel. J'initialise ensuite Git pour pouvoir pousser et suivre mes modifications au fur et à mesure du développement de l'application.

Mise en place d'un template Bootstrap

Ma première étape fût de mettre en place un front qui se verra pourvu de mes futures informations traitées. La tâche n'a pas été très difficile car il existe des exemples de template sur Bootstrap. Je l'ai donc adapté à mes besoins.

Le parser

J'ai commencé par m'intéresser au module Nltk qui est un parser visiblement assez utilisé. Mais j'ai finalement opté pour un parser à base de *stopwords*. Il est certes un peu naïf avec des requêtes basiques, mais pour des phrases plus complexes, le parser s'emballe un peu. Il y a une bonne perspective d'évolution pour cette fonctionnalité.

L'appel AJAX

Ce fût la partie la plus difficile car je ne comprenais pas comment fonctionner le système d'envoi et de réception des données.. J'ai cherché pendant plusieurs heures une solution qui fonctionne. Et finalement j'ai bien compris le principe d'AJAX qui envoi des données puis effectue une action sur le *callback*.

API Gmap

J'ai eu quelques difficultés concernant l'obtention d'une clé API. Je ne savais pas que depuis un moment Google demandait le numéro de carte bancaire pour pouvoir obtenir cette dernière.

API Wikipedia

Pas de difficulté sur ce point, la documentation est très complète. On retrouve facilement les informations en envoyant un mot clé dans le paramètre search de l'url. Le fichier Json reçu est très facilement exploitable et contient toutes les informations pour nos besoins.

Déploiement sur Heroku

La difficulté majeure du déploiement sur Heroku est que Flask n'est pas acceptable par défaut sur Heroku. Il a donc fallu faire des manipulations spécifiques pour accueillir Flask sur le serveur web et le déployer. L'application est donc en ligne à cette adresse :

(rickpybot.herokuapp.com)

Respect de la PEP8

Pour que mon code soit propre, il a fallu que j'utilise des outils tels que pylint, black et les conventions de la PEP8.

J'avais peu travaillé sur ces aspects lors des précédents projets. Lorsque j'ai lancé la première fois le module pylint sur mon code qui contient toutes mes fonctions, la note était de -2,10 sur 10. J'ai donc repris toutes les bonnes pratiques afin d'obtenir la note de 10/10. Cela inclut des lignes inférieures ou égales à 100 caractères, des docstrings à chaque fonctions, des conventions de nommages, la suppression d'import non utilisé, etc

J'ai formaté une partie automatique avec un module qui s'appelle black. Cela a m'a fait gagné un peu de temps. J'ai également utilisé les règles et l'autoformat de visual studio code.