Maxime J.



##### **Projet 7 Openclassrooms : Créez GrandPy Bot, le papy-robot 🤖 👴**

**Livrables :**

* Document texte expliquant la démarche choisie, les difficultés rencontrées et les solutions trouvées. Le document doit être en format pdf et ne pas excéder 2 pages A4. Il peut être rédigé en anglais ou en français, au choix, mais prenez bien en considération que les fautes d’orthographe et de grammaire seront évaluées !
* Adresse du site déployé pour consulter votre projet "en vrai" et poser des questions à GrandPy (les mentors aussi ont le droit de s'amuser !) : <https://grandpybotte.herokuapp.com/>
* Code source hébergé sur Github : <https://github.com/Maximedu13/Cr-ez-GrandPy-Bot-le-papy-robot>
* Tableau Trello ou Pivotal Tracker contenant les user stories : <https://trello.com/b/bpszZ58s/grandpy-bot>

1. **Generalities**

The aim of this project is to create a website providing access to information related to a question raised by the user. He enters a question and GrandPyBot answers him displaying:

* An extract from the Wikipedia page concerning the address of the place retrieved via Wikipedia API.
* A map retrieved via Google Maps API pointing to the place the user is looking for.

Nothing is saved and the user sends his question by pressing enter and the answer is displayed directly on the screen, without reloading the page.

The program is coded in HTML, CSS, JavaScript and Python on Visual Studio Code 1.33.1. The front-end is developed thanks to the Framework Bootstrap and the back-end thanks to the Framework Flask.

This program is hosted on GitHub, here is the link : <https://github.com/Maximedu13/Cr-ez-GrandPy-Bot-le-papy-robot> and on Heroku server, here is the link : <https://grandpybotte.herokuapp.com/>

1. **Content of the project**

Flask gives more freedom (than Django for example) to allow each developer to organize himself as he wishes.

The project contains the following programs and files:

* a repository GrandPyBot : contains the application itself.
* a repository tests : this folder contains the unit tests.
* README.md : a file containing information about other files in the same folder.
* config.py :
* requirements.txt : containing a list of commands for pip that installs the required versions of dependent packages. The program libraries used are:
* run.py : this file is used to run the Flask application.

1. **Approach chosen**

It was requested to adopt the Doc Driven Development approach, in other words to, respectively, determine the list of features to provide, write the complete documentation, write the code, check if the doc is well respected, and finally iterate.

1. **Algorithms choice**

In this part, is developed some of algorithms used in the program.

**The parsing algorithm:**

It's useful to break long sentences into words, which will be analyzed to keep only relevant key-words (an address for example).

First, we need to pick up a list of stop words from this link : <https://github.com/6/stopwords-json/blob/master/dist/fr.json>

Secondly, we have to think about

**The algorithm:**

1. **Difficulties encountered and solutions found**

Many miscoding (errors) were obstacles to the proper functioning of the program:

❌ {"error\_message" : "You have exceeded your daily request quota for this API. If you did not set a custom daily request quota, verify your project has an active billing account: http://g.co/dev/maps-no-account",

"results" : [],

"status" : "OVER\_QUERY\_LIMIT" }

❓ Almost every Google API has a daily free quota. For Google Places API it's 2500 requests per day and 10 requests per second.   
✅ Enable billing.

❌ ‘Application Error:  
An error occurred in the application and your page could not be served. Please try again in a few moments.  
If you are the application owner, check your logs for details.’

❓ gunicorn is not using the correct port. Heroku assigns a port for the application.   
✅ First, create a PROCFILE file and fill it with this instruction :

web: gunicorn GrandPyBot:app

Secondly, add gunicorn==19.9.0 to requirements.txt

Thirdly,

1. **Possible improvements**

To improve the project, it would have been possible to have recourse to a JavaScript framework such as Vue.js, as we used   
respectively the frameworks Bootstrap for HTML/CSS, and Flask for Python. The advantage of Vue.js is to be easy to understand and to develop Applications. It facilitates the developers to integrate with the existing applications.

It will still be possible to improve the parsing algorithm. In fact, there are still special cases that do not return any response. For example, the search on the name of the street will not give anything for the streets containing a date (ex: Rue du 11 Novembre 1918) because the algorithm removes each numeral.

These improvement tracks could be the subject of the project 11 on Openclassrooms “Improve an existing project in Python”.