Foodie-Web Service

By Advith Chegu (ac1771)

Q: A short paragraph or two describing what you accomplished.

A: My foodie web service is a REST Api built using python's flask library along with flask's swagger documentation library used to test my service. The docs can be reaches by entering `localhost:5000/docs` in a url bar when you run the service locally.

The service takes in your address as an argument and uses the Geocod.io api to pinpoint the coordinates of your address. It returns an error status code if the address was not found or if the client enters a malformed address.

Next, using the coordinates, Foodie-Web uses Yelp's Fusion api to find the nearest 20 businesses and returns the result as a JSON. Because the goal of the service is just to give the client the name, address, and rating of the restaurant, the service parses the returned JSON and forms its own object to return.

```
# service returns a list of business models
BUSINESS_MODEL = api.model("Business", {
    'name': fields.String(required=True),
    'address': fields.String(required=True),
    'rating': fields.Float(required=True),
})
```

Q: A short paragraph or two describing any issues you may have encountered and how you think you could have solved them. If you didn’t have any simply state that you didn’t have any issues.

A: I used flask every day in my internship which is mostly the reason why I did not encounter any issues. However for someone unfamiliar with the space I can see why building a REST Api might seem a bit complicated.

Q: A short description on the libraries/packages you used

A: All the packages I used can be found in my pipfile, I'll go over the most important ones for this project

flask: Flask is a web framework written in python which allows developers to create and test their REST Apis very quickly.

python-dotenv: This library allows developers to easily read secrets from their `.env` file. I used this to store the keys used for yelp and geocod so that I wouldn't accidentally expose the keys when being pushed to a remote repository.

requests: This library allows developers to easily make requests to any publicly exposed api/webpage on the internet. I used it to fetch the Geocod and Yelp data.

Q: A short description on how to start your web service

A: The first thing that you must do is install the packages. You can do this by first running `pipenv shell` then running `pipenv install` which will install all of the packages in the Pipfile file in this folder.

(If you're using VSCode) I've included the .vscode file, you should simply be able to go to the debugging panel and press the green play button which says Python: Flask.

(If you're not) First cd to this folder. Next you need to tell flask which is the main file. You can do this by copying the following command into the terminal: `export FLASK_APP=main.py`. Next, you can simply run the following command in your terminal after you `python -m flask run` and the app should be visible if you hit `localhost:5000/restaurants`

An example query would be as follows:

```
curl -X 'GET' \
  'http://localhost:5000/restaurants?
Address=40%20College%20Ave%2C%20New%20Brunswick%2C%20NJ%2008901' \
  -H 'accept: application/json'
```

Q: A short description of where you placed your Geocod.io / Yelp API key within your code. (This way I can replace it with my own if I need to)

A: Just create a .env file in this folder with the following information:

```
geocode-key=
yelp-client-id=
yelp-key=
```