JBT Pizza Delivery

This feature is created to calculate the delivery routes for a small pizza chain that recently purchased a drone to deliver their pizzas. Drone range is 25 miles and you can configure that when you create drone object.

I have used Flask to create the endpoints.

VirtualEnv

First create the virtualenv using below commands.

Install the virtualenv
pip install virtualenv

Initialize the venv in your local dir
virtualenv venv

Activate the virtualenv
source venv/bin/activate

Now you are good to install all of your pip packages.
.//venv/bin/pip install -r requirements.txt

Now the app is ready to test. This app is created in Flask and it has 1 API Endpoint

Run this command to start the app

flask --app src/app run

Now your app is running on port 5000. If you run through docker, then port will be 35000. I have included docker-compose file as well.

This code test assumes pizza shop address is = "6138 Bollinger Rd, San Jose, CA 95129" so all the drone delivery will happen from this pizza shop

API Endpoints

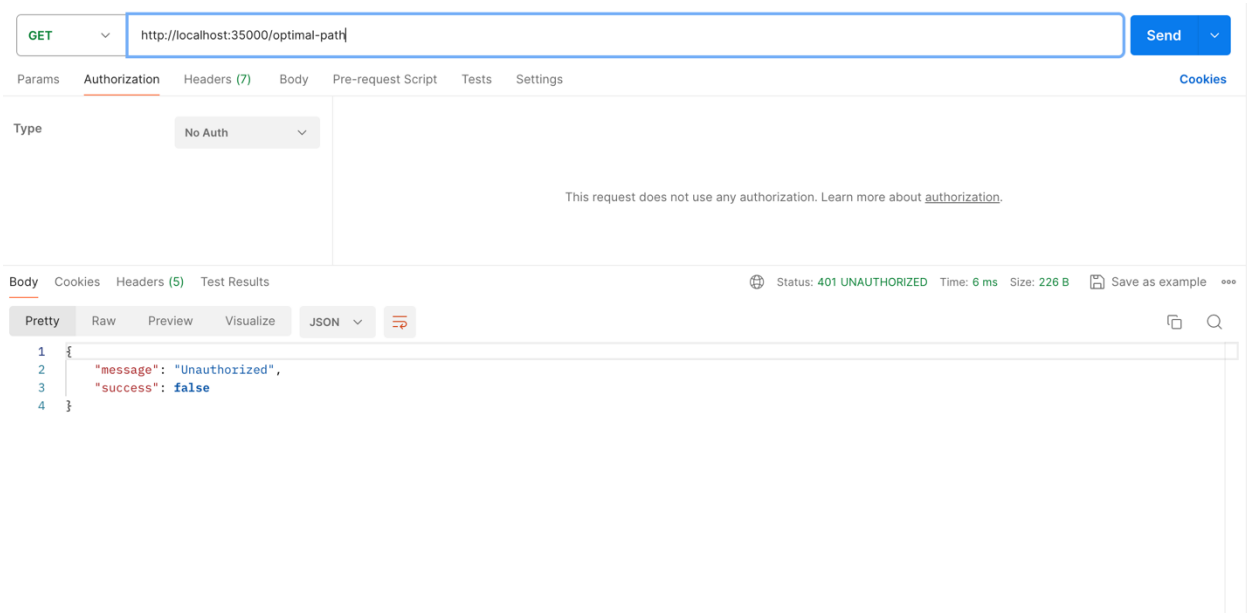
1. First endpoint is to calculate the optimal path for given addresses in orders.csv file. I have used Geopy library to calculate the optimal path.

First, I fetched the first address from the CSV file and then using the Geopy library, I fetched the nearest address from rest of the addresses, and after fetching that second address, I repeated the processes and I kept getting nearby addresses.

GET {{server}}//optimal-path

Make sure you first authenticate and get the jwt token. Pass this jwt token to all the endpoints in this app.

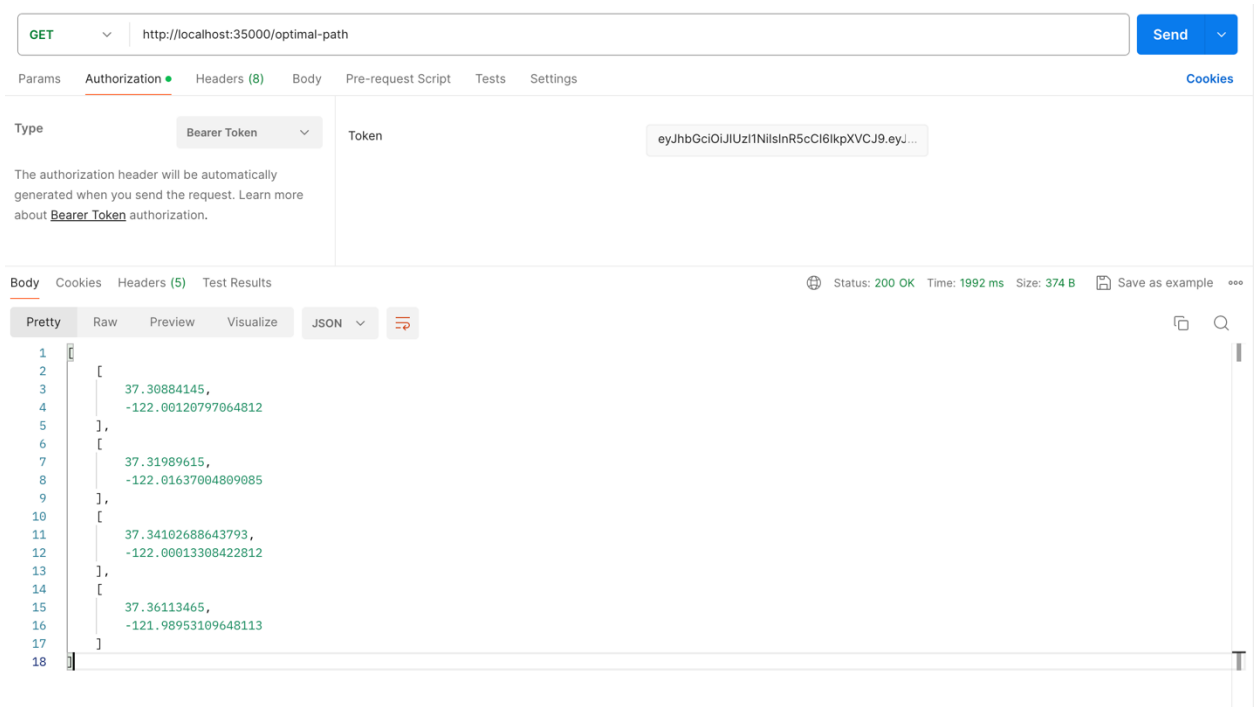
This is the error it will report if you are unauthorized



This return the path the drone should take to deliver


```
[
  [
    37.30884145,
    -122.00120797064812
  ],
  [
    37.31989615,
    -122.01637004809085
  ],
```

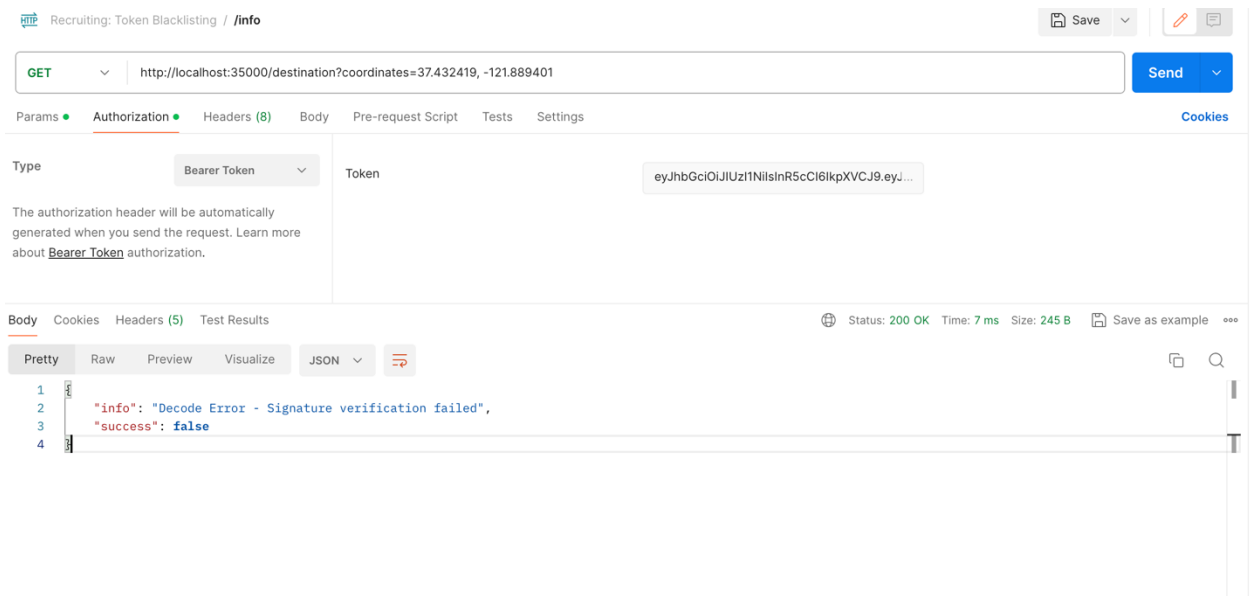
```
  [
    37.34102688643793,
    -122.00013308422812
  ],
  [
    37.36113465,
    -121.98953109648113
  ]
]
```

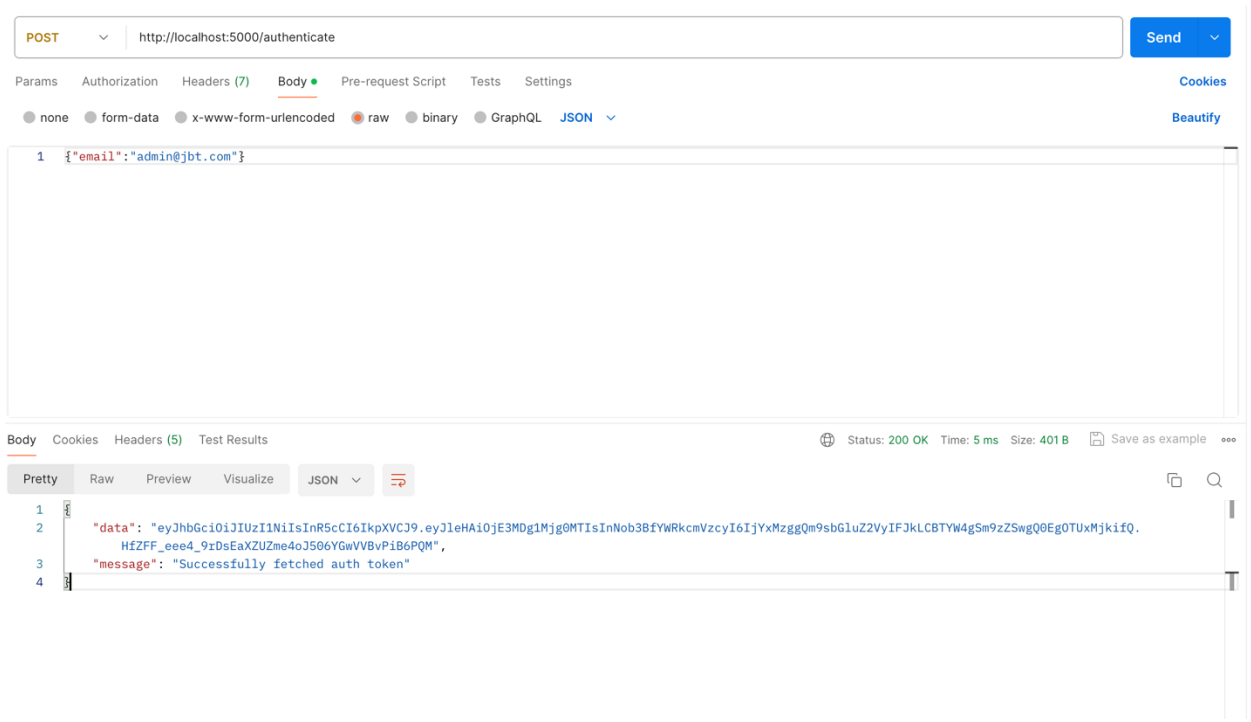
2. This endpoint provides the authenticate functionality. Currently only user "admin@jbtc.com" can ask for jwt token using authentication endpoint.

This endpoint will return the jwt token which is valid for 1 hour.

If someone else other than admin@jbtc.com try to use the endpoint using token, it will throw “Token decode” error.



POST {{server}}/authenticate




```
{
  "data": "token",
  "message": "Successfully fetched auth token"
}
```

3.

POST {{server}}/destination?coordinates=37.461449, -121.912258

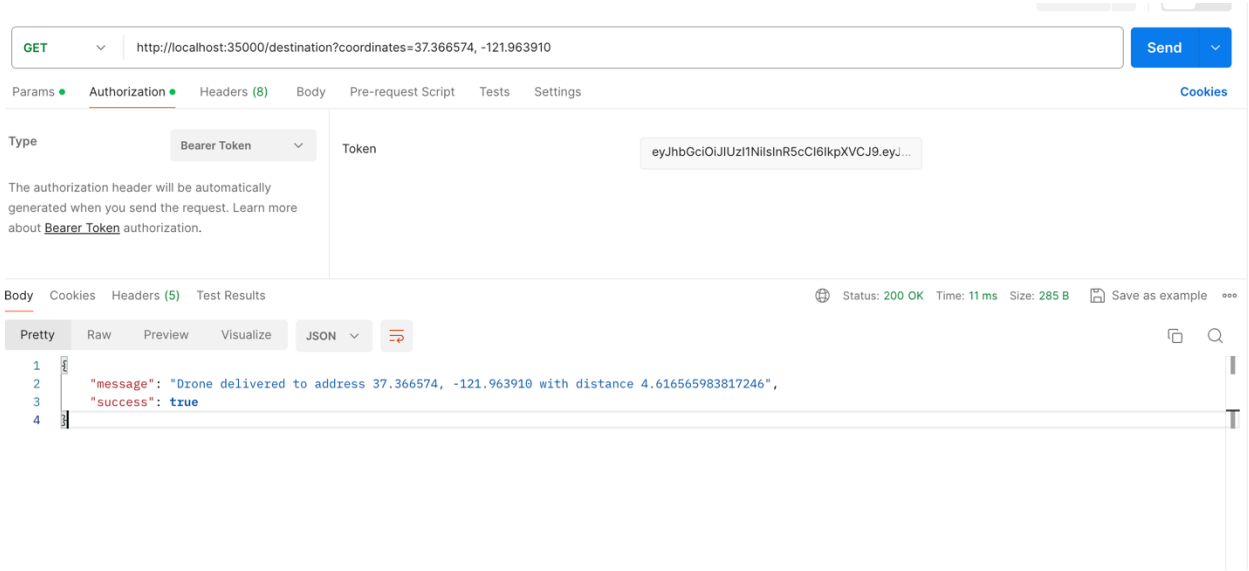
This endpoint return "success" : True if drone can deliver to given address, otherwise it returns False if drone is out of range for that address.

This is the formula it uses to deliver the next destination. First check the existing range drone has remaining out of 25 miles. Next check the distance from current address to next destination plus distance back to the pizza shop from that destination. If it is within the range currently drone has then it is ok to deliver otherwise endpoint throws message “it is out of the range”

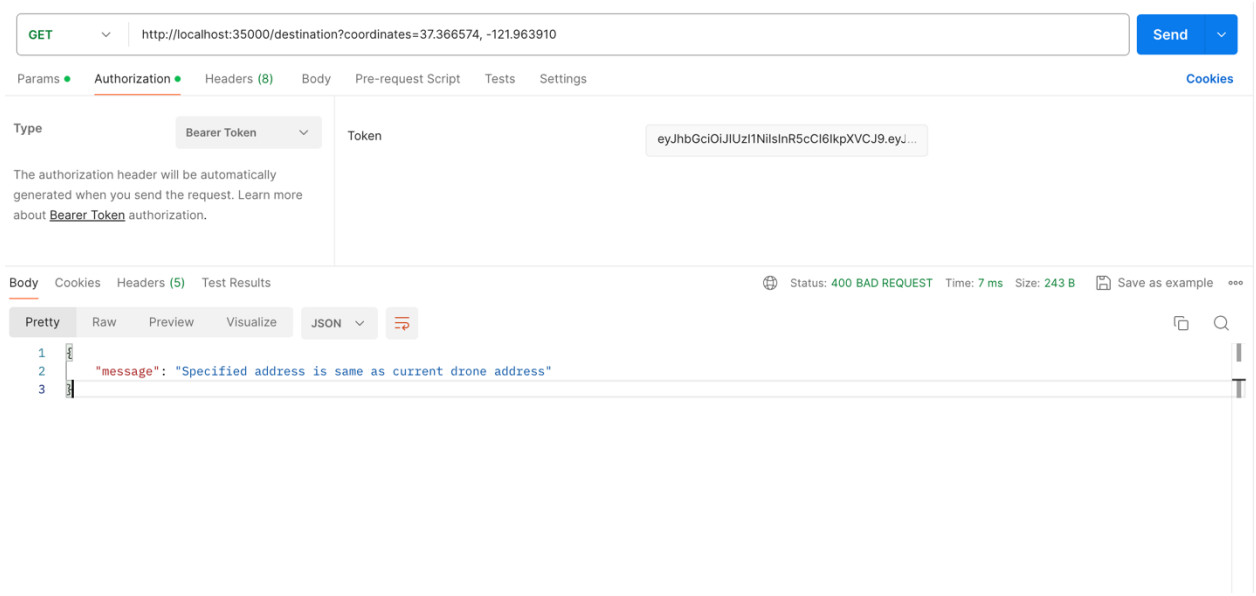
{
  "message": "Drone delivered to address 37.461449, -121.912258 with distance 11.710116663456743",
  "success": true
}

If drone is out of range, it will return 500 with message

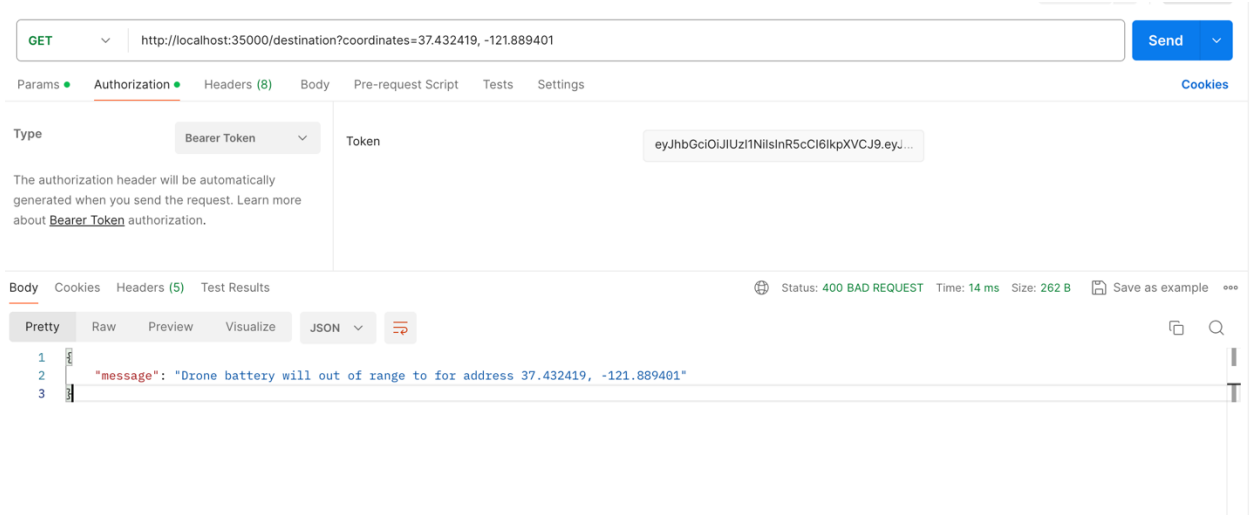
{
  "message": "Drone battery will out of range to for address 37.366574, -121.963910"
}



If you specify same delivery address again, it will report error stating, it is same address.



Drone will keep delivering until it is out of range (25 miles).



Tests

I have included 2 test cases one with success 200 and another with failure (no param) 404.

To run the test, execute below command

pytest

```
============================================================ test session starts
============================================================
platform darwin -- Python 3.10.9, pytest-8.0.1, pluggy-1.4.0
rootdir: /JBT-Test/jbt-test
collected 2 items

tests/test_endpoint.py ..                                                [100%]

============================================================ 2 passed in 0.67s
============================================================
```

Docker

I have included docker-compose file in case you want to create Docker image.
Run this command to create the docker image

docker-compose up

Questions/Concerns:

I used KISS (Keep it Simple,Stupid) principle to create this app. I created utils functions in separate file. If you have any questions or concerns please reach out to me keyurpatel80@gmail.com