WoltBot

Introduction

Hey there! It's raining outside and I want to order pizza but my restaurant is temporarly closed in Wolt! I haven't eatten all day, and my duda is killing me.
I need your help to build a Telegram
WoltBot that'll check for me if the restaurant is back online.

Stack

BE - FastAPI, python-telegram-bot

Telegram - Bots creation API
Follow the expected software best practices for each of the above.

Phase 1

Create a new bot on the Telegram platform. Choose a cool name for the bot and a meaningful username. Make sure to save the newly generate API token!

Phase 2

1. Setup locally your python project.
    a. Project should start from a `main.py` file.
    b. Project should use the python-telegram-bot library.

2. Your code initially should do the following
    a. When a user interacts with the bot by sending `/start` → You'll send him back the following:

    ```
    Hey <Telegram username>!
    ```

    b. When a user sends his current location → Save the lattitude / longitue in global vars and send him back the following:
    `Your are here: <longitude>/<latitude>! I saved it.`

Phase 3

1. Set the following restarutant mapping:

```
RESTAURANT = {
    "Fat Cow": "fat_cow",
        "Greco": "deli-by-greco",
        "Silly Kid": "silly-kid",
        "Souplier": "souplier",
}
```

2. When a user interacts with the bot by sending `/start` → You'll send him back the following output:

```
Hey <Telegram username>! Which restaurant should we check for you?
1. Fat Cow
2. Greco
3. Silly Kid
4. Souplier
```

3. Upon selecting one of the above restaurants, use the following url to check the availabillity of the restaurant:

    https://consumer-api.wolt.com/order-xp/web/v1/venue/slug/<RESTAURANT_ID>/dynamic/
    Parse the returned json and send back to the user
    `Open!` or `Closed!` based on what you found.

Phase 4

1. When a user interacts with the bot by sending `/start` → You'll send him back the following output:

```
Hey <Telegram username>! Which restaurant should we check for you?
1. Fat Cow
2. Greco
3. Silly Kid
4. Souplier
```

2. Upon selection, send the following and request the user to share his current location:

```
Great! What is your current location?
```

3. Upon receiving a valid location, Save the lattitude / longitue in global vars and use the following url to check the availabillity of the restaurnt in respect to his current location:

    https://consumer-api.wolt.com/order-xp/web/v1/venue/slug/<RESTAURANT_ID>/dynamic/?lat=<LATITUDE>&lon=<LATITUDE>
    Parse the returned json and send back to the user
    `Open!` or `Closed!` based on what you found.

Phase 5

Upon implementing phase 4, keep on checking the restaurant for its availabillity based on the user’s location, every 30 seconds, until one of the following happens:

1. Restaurant is open.
2. 10 minutes has passed since the request was submitted by the user.