As of 8/9/18

Application ID: CI03280888

To Set Up Locally:

Requires:

Collaborator in Cafebot App in <https://api.slack.com/apps>

Python3.7, pipenv, ngrok

Steps:

In terminal, ‘ngrok http 5000’

In Slack/Apps (https://api.slack.com/apps), manually update:

* Event Subscriptions request URL to [https ngrok addr]/listening
* Interactive Components URL to [https ngrok addr]/interactive

In project directory, run ‘pipenv run python app.py’

(after installing pipfile.lock dependencies)

Navigate to localhost:5000/install and click through to add app to Slack and allow permissions for Cafebot to DM users (will need to be done each time app.py is run / updated)

(Note – ‘rm user\_settings.db ; cat create\_db.sql | sqlite3 user\_settings.db’ will need to be run every time db code is changed as well)

To Use Locally:

Requires:

Access to CafeBot app info

Postman / anything to make authenticated post requests

Info for messages, such as channel and user ID

Steps:

Get Postman OAuth 2.0 token (fill out client ID, client secret, authorize)

Make a POST request to [https ngrok addr]/listening

team\_join event (for when a user joins a workspace) triggers welcome message:

{

"token": "pVa2ph8ubn4agffdKxnEGHPB",

"team\_id": "T04SM6T1Z",

"api\_app\_id": "AC2PWMG23",

"event": {

"type": "team\_join",

"user": "UC1S8K38D",

"ts": "1533309313.000443",

"channel": "DC4BC3D4Z",

"event\_ts": "1533309313.000443",

"channel\_type": "im"

}

}

* Will need to change user and channel fields to your own
* Click through to choose settings and save those to the in-memory db

For the menu sending:

Currently scheduled to send once a day at 7AM

Can modify code to test so that upon app startup, function that calls send\_menu() in app.py sleeps (while you manually allow permission and then if you want, choose settings), then runs once, before sleeping again (for however long)

TODO for the Future:

For Deployment:

- Need to use waitress / some other WSGI server instead of Flask

- Update all of Slack App settings to server settings

- There’s a whole process for deploying Slack apps to the Target workspace

Cleanup

- Update to not use verification token (outdated) and use signing secret instead

- Put message json files in a messages folder

- Honestly not sure whether some of the functionality implemented in bot.py should be in bot.py or a separate module

- Probably also a better way to structure sending messages instead of using a combination of the message.py module, message\_user(), and self.client.api\_call()

- Currently making a second context for db in app.py, should find less hacky way

- Make CafeSettings and PriceSettings use a foreign key in UserIds instead

Features:

- Check if slackeventsapi library is up to date, could use that instead (as of now the Github repo is but the version on Pypi isn’t)

- Add functionality where users can ask Cafebot about nutritional info, Cafebot detects based on keywords and returns said info

- Add ability to change chosen settings