
COMP 523: Admin Manual

10th December 2023

OVERVIEW

Ego Green is a mobile app that is for those who are interested in monitoring their environmental practices and contribution to sustainability. On its initial iteration, the app will monitor daily sustainability practices by making estimates on carbon footprint – based on input data. Through the use of user-inputted receipt data from grocery stores and Open AI, users will receive a carbon emission output so that they can get a better understanding of their environmental impact and start taking steps to change it. With this app, we aim to give users relevant information and the ability to increase their sustainability practices so that they can create a more sustainable world for all.

SET UP AND INSTALLATION

There are two separate installation processes for the Android and iOS parts of the applications.

These instructions are done based on the assumption that Nativescript is being installed on macOS.

1. iOS
 - a. Install HomeBrew
 - i. Refer to this website for further instructions: <https://brew.sh/>
 - b. Install Node
 - i. `nvm install node` [can use n or npm or any other node version manager]
 - ii. Verify installation:
 1. `node -v`
 2. `npm -v`
 - c. Install Xcode
 - i. Ensure that the command lines tool has been selected within Xcode settings
 - d. Install Ruby
 - i. `brew install ruby@2.7`
 - ii. `brew link ruby@2.7`
 - e. After installing ruby, there should be instructions that you can type into the command line to directly add this path to the configuration file

-
- f. Add in the path manually (if the previous step did not work)
 1. Open up the configuration file relevant to your terminal window type (bash/zsh)
 2. Add the ruby path to the file
 3. It should look similar to:
 - a. `export PATH=/opt/homebrew/lib/ruby/gems/2.7.0/bin:$PATH`
 - b. `export PATH=/usr/local/lib/ruby/gems/2.7.0/bin:$PATH`
 - g. Install cocoapods and xcodeproj
 - i. `sudo gem install cocoapods`
 - ii. `sudo gem install xcodeproj`
 - h. Ensure installation was successful
 - i. Run `which pod`
 - ii. If you encounter errors, make sure that the paths in the profile files are correct and have been updated
 - i. Install Python3
 - i. `brew install python`
 - ii. `python3 --version` [ensure that the installation was successful]
 - j. Update pip and six
 - i. `python3 -m pip install --upgrade pip`
 - ii. `python3 -m pip install six`
 - k. Install the NativeScript CLI globally
 - i. `npm install -g nativescript`
 - l. Ensure successful installation
 - i. `ns doctor ios`
2. Android
- a. If you have completed the iOS process, you can skip some of these steps
 - b. Install Homebrew (reference earlier instructions)
 - c. Install Node (reference earlier instructions)
 - d. Install a JDK
 - i. `brew tap homebrew/cask-versions`
 - ii. `brew install --cask temurin11`
 - e. Verify installation with
 - i. `javac --version`
 - f. Add JAVA_HOME environment variable to the shell profile (bash or zsh profile depending on your terminal type)
 - i. `export JAVA_HOME=$(/usr/libexec/java_home -v"11")`
 - g. Install Android Studio
 - h. Add ANDROID_HOME and PATH environment variables to the shell profile (bash or zsh depending on the terminal)

-
- i. export ANDROID_HOME=\$HOME/Library/Android/sdk
 - ii. export PATH=\$PATH:\$ANDROID_HOME/platform-tools
 - i. Install NativeScript CLI (reference earlier instructions)
 - j. Ensure successful installation
 - i. `ns doctor android`

Refer to this website for any further questions: <https://docs.nativescript.org/setup/macOS>

CLONE REPOSITORY

1. Open the GitHub repository:
<https://github.com/COMP-523-Sustainability-Tracker/egogreen>
2. Click on the Code Button (green button with dropdown)
3. Copy the HTTPS link
4. Open a new Terminal window
5. Navigate to a folder where you'd like to store the project
6. Run `git clone <HTTPS LINK>`
7. You should now be able to navigate to the egogreen folder and see the Nativescript project
8. We recommend using a text editor such as VSCode in order to edit/view the code

RUN APPLICATION

EMULATOR

1. Open a new terminal
2. Run `ns run ios` or `ns run android`
3. Interact with the application using the emulator that will open shortly

PHYSICAL DEVICE

1. Download the NativeScript Preview Application on your physical device
2. Run `ns preview` (or connected you developer mode enabled phone by USB and verify that nativeScript sees with the command `ns devices`)
3. This will generate a QR Code that you can scan with your physical device's camera
4. This should open up the application on your device
5. You should now be able to interact with the application

API REFERENCES

-
1. We upload the receipt image taken with a phone's camera via POST our NodeJS server running on heroku.com once uploaded the image input data is sent through 3 API processing steps sequentially for CO2e classifying and assessment.
 2. Taggun - <https://developers.taggun.io/reference/scan-file-verbose>
 - a. This API extracts text including the line items from the receipt image into JSON data. (OCR is always a possible point of failure)
 2. OpenAI - <https://platform.openai.com/docs/api-reference>
 - a. We then process each line item through a 4 step hierarchical classification scheme using UNSPSC (<https://finance.cornell.edu/procurement/buyers/unspsc>) stored on the NodeJS instance as an SQLite file DB.
 - i. Each step consists of asking OpenAI which category is the best match for the line item and then repeating with the matched category's children. (Line items, especially concise description are sometime difficult for openAI and humans to decipher, the 4 step process provides the best effort category and falls back to a default of food purchases.)
 3. Bend.Green - <https://docs.bend.green/>
 - a. Once we have the most detailed UNSPSC code we can match for a line item the dollar spent amount and UNSPSC code is sent to bend green for a CO2eg assessment.
 4. The results of the 3 steps are then saved to a User's receipt collection in FireBase Firestore.

AUTHENTICATION FILES

Sustainable.io already has the API keys and cost information for OpenAI and Taggun.io, Bend.green's CTO Thomas Moore provided free access to the Bend.green api while the app has low volume and will be introduced to our clients by email at handoff time.

All API and DB keys for the node instance are stored in a .env file for local development and as environment variables in the heroku.com instance.

The app only communicates directly with Firebase and the nodeJS instance. The firebase keys are available in the Firebase console and will be emailed along with the .env at handoff.