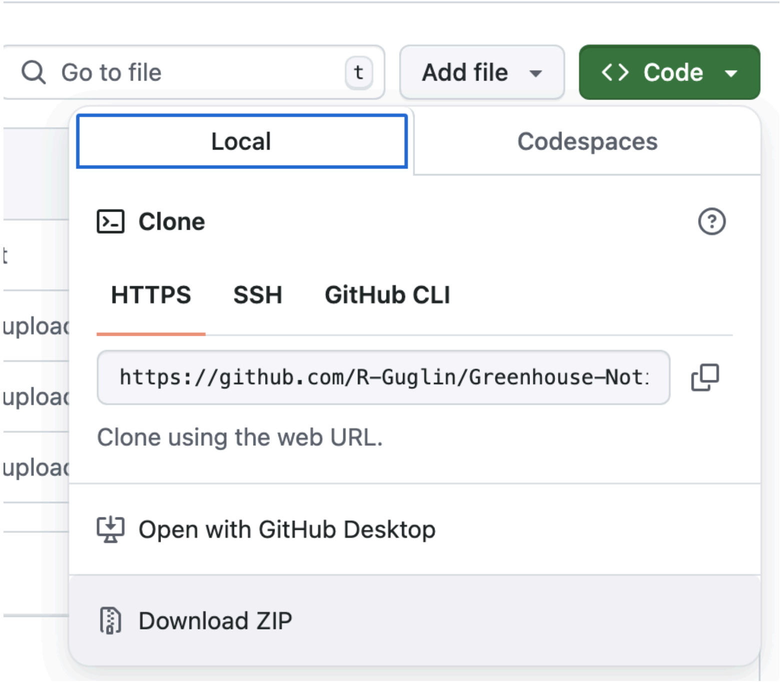
Installation

1. Go to https://github.com/R-Guglin/Greenhouse-Notion-API.
2. Click the green "Code" button, then select "Download ZIP."



https://github.com/R-Guglin/Greenhouse-Not:

Clone using the web URL.

3. You should now have a ZIP file in your designated downloads folder. Double click this file to expand its contents.
4. You should see four files: a README and another ZIP called dist.zip — expand dist.zip into its contents as well.
5. Finally, you should end up with a folder called **dist** which contains two items: a folder called **main**, and an *executable* called **main**. The executable is our integration app.

6. If you want, you can drag the executable into your Applications folder, desktop, or wherever else you choose. It doesn't need to be in the same location as the Main folder. But make sure not to delete that folder, since it contains all the libraries and other dependencies that our code runs on.
7. For extra convenience, try making a folder called that contains all of the things you'll need to run the program:
    a. The executable
    b. Any .csv files downloaded from Google forms which will be fed into the program.

Getting the CSV

1. Go to forms.google.com and edit the form you are hoping to pull information from.
2. In the "responses" tab, navigate to the form's linked Google Sheet.
3. Make a copy of this Sheet, and delete any rows corresponding to people you don't want to add to the directory. You do not need to delete any columns. You also don't need to delete duplicates, as the program will check the directory for existing people before adding their information again.
4. **Important: Renaming Columns**
    a. The way my program works is to scan through a CSV for columns with certain titles, like "Location" or "Name." Google Forms automatically names its columns with the full title of the question. Therefore, some of the columns will need to be renamed. **Each relevant column must be given a name that exactly matches one of the attributes in the list below, including capitalization**. You do not need to rename columns that aren't related to these attributes.
    b. The full list of attributes that are currently supported:
        i. **Names**: this is slightly more complicated due to a disparity in how names were collected in the two different Google forms.
            1. If this is a Community upload **or a Cohort upload with a timestamp on or later than 8/22/2024**, the *single* column storing names should be relabeled "Name".

2. If this is a Cohort upload **and the rows you're adding have a timestamp from before 8/22/2024**, then there are *two* different columns, one for the first name and one for the surname. The first should be relabeled "First Name" and the second should be "Last Name."

ii. Email
iii. LinkedIn
iv. Ask – what they are hoping to get out of the program/community.
v. Give – what they contribute to the program/community.
vi. Problem – the climate problem(s) they're working on.
vii. Website — a URL to their website if they have one.

5. Once you have renamed the necessary columns and removed any rows you don't want to keep, go to File → Download → Comma-Separated Values (CSV) and save the file.
6. (Optional) Drag the CSV file into the folder you created with your executable.

Running

1. The first time you try to run the executable, Apple will block it because it can't be scanned.
   a. Open System Settings → Privacy and Security
   b. Scroll down to near the bottom, and there should be a note saying that the system blocked an app called "main" from opening. Click "Open Anyway" and enter your password if necessary.
   c. The app should open.
2. Now, just enter the integration key and the page ID of the database you're trying to modify.
   a. If you can't find the database ID or you get an error saying it isn't recognized, you can try the following:
      i. In your browser, navigate to any page inside the database.
      ii. Copy the page ID from this page: the 32-character string of digits and numbers at the end of the page's URL.

        iii. Paste that page ID into the app and click “Fetch Parent Database.” It should return the ID of the database you’re trying to modify. (For more info on why this is included, read the comments in integration.py)

3. Once all the necessary information is filled in, click “Add to Database”.

Notes/Possible Issues

1. When the program checks the directory for duplicates, it is looking for *exact* copies of someone’s email address. If someone didn’t provide an email, this won’t work.