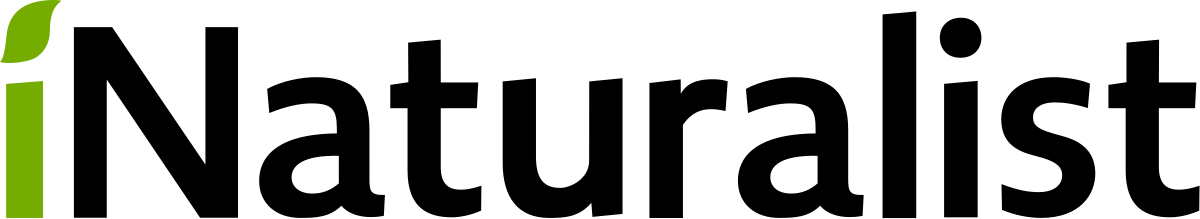
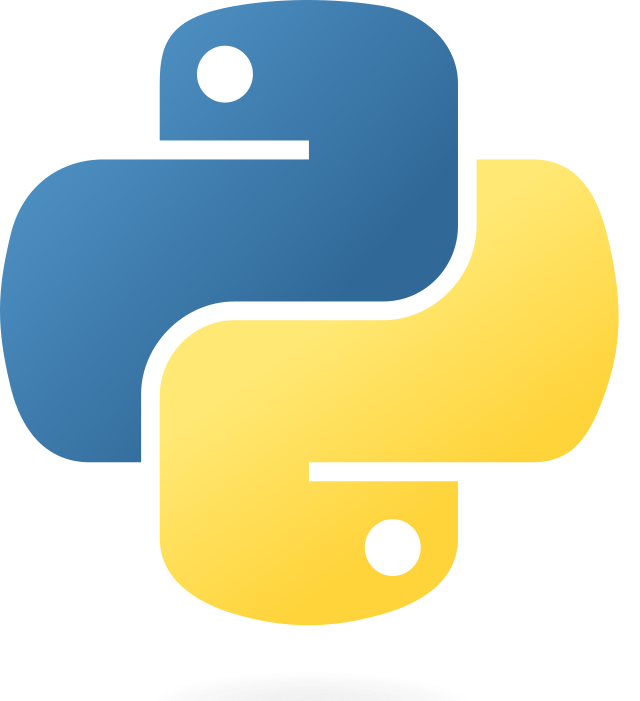
Maintaining Biodiversity and Temperature Database Manual

A how-to guide for setting up your own Automated Biodiversity and Temperature Database









Written as Supplementary Material to an Interactive Qualifying Project in partial fulfillment of a Bachelors of Science Degree from Worcester Polytechnic Institute.

24 January, 2023

Written by:

Lily Bromberger and Eric Schuman

["Google Developers text logo"](https://developers.google.com/) by Google is in the [Public Domain, CC0](http://creativecommons.org/publicdomain/zero/1.0/)

["Naturalist text logo"](https://www.inaturalist.org/) by iNaturalist (California Academy of Sciences & National Geographic) is in the [Public Domain, CC0](http://creativecommons.org/publicdomain/zero/1.0/)

["OpenWeather-Master-Logo RGB"](https://openweathermap.org/) by Openweather is licensed under [CC BY-SA 4.0](http://creativecommons.org/licenses/by-sa/4.0)

["Python-logo-notext"](http://www.python.org/) by [Python](http://www.python.org/) is in the [Public Domain, CC0](http://creativecommons.org/publicdomain/zero/1.0/)

["QGIS logo minimal"](https://www.qgis.org/en/site/getinvolved/styleguide.html) by QGIS software is licensed under [CC BY-SA 3.0](http://creativecommons.org/licenses/by-sa/3.0)

# Table of Contents

[Table of Contents](#_eeg1vswvczhn) 2

[Notes and How to Use This Manual](#_1ehafpovh2i8) 3

[Section 1: Setting Up Automation Of iNaturalist and OpenWeather Data Collection and On A Linux Server](#_xogrdh2dbhk0) 5

[Section 2: Setting Up Manual Data Collection from iNaturalist](#_fbuicc4g1anz) 12

[Section 3: Setting Up Manual Data Collection from OpenWeather](#_uig3gtur8li7) 14

[Section 4: Managing Heat Gun Data](#_mii9zfq0e55y) 15

[Section 5: Adding GeoJSON Files to Reflect the Coordinate Boundaries of a Specific Location](#_z3bs8cf8dnvy) 16

[Section 6: Creating new Google Sheets Credentials](#_y1gjeu703lt1) 20

[Section 7: Database Column Descriptions](#_9su6xk3i8kpq) 24

[1. Biodiversity Data](#_mj161smc6qul) 24

[2. Melbourne Airport Data](#_hopyntjcy8k5) 26

[3. Heat Gun Data](#_ijvqa9toumyr) 27

[4. Temperature Sensor Data](#_1tn28obq6hqw) 30

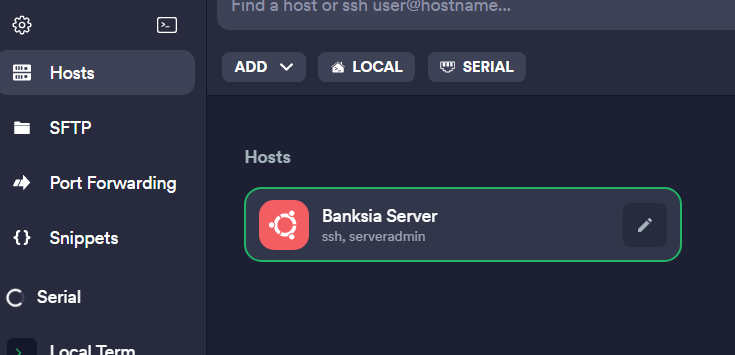
[Sources](#_d777ijp29kod) 31

# 

# Notes and How to Use This Manual

* This manual should be used to help with the upkeep of the “Biodiversity and Temperature Database” ([Linked Here](https://docs.google.com/spreadsheets/d/16o1ZaQYOkmDtdOdNgwxmLvz8LCSCXItBTcIfWj7PB5M/edit#gid=292870469)) created for Banksia Gardens Community Service to track biodiversity and temperature on their property over time.
  + While the manual and database were designed to meet the specific needs of Banksia Gardens Community Service, they can also serve as a model for replicating the system or any of its individual elements
* The database has four parts all split into individual worksheets:
  + **Biodiversity Data** - tracks all biodiversity observations submitted to the iNaturalist project Banksia Gardens Biodiversity ([Linked Here](https://www.inaturalist.org/observations?project_id=154294))
  + **Melbourne Airport Data** - tracks temperature recordings every hour at the Melbourne Airport
  + **Heat Gun Data** - tracks manual heat gun recordings around Banksia Gardens’ property
  + **Temperature Sensor Data** - tracks temperature sensor recordings around Banksia Gardens’ property
* How to Use This Manual (Directions For Banksia Gardens updated as of 17/2/2023)
  + **Biodiversity Data**
    - Currently this worksheet is being automated by Banksia Gardens’ linux server by following the directions in Section 1.
      * The server is set to automatically refresh the current worksheet. If you would like to create a new worksheet, you must manually do so using Section 2
    - This worksheet should be monitored frequently, and if there are problems with the data upload you can manually update the sheet or create a new one by following Section 2.
      * Since the iNaturalist site has its own database, all of the biodiversity data is backed up. If necessary, all data can be recovered by deleting the current biodiversity sheet and creating a new one.
    - Whenever a new garden is added or is expanded, follow Section 5 to add or update a “region” in the code. This will edit the biodiversity.py file, so make sure that you replace the old file with the new file in the linux server if you are running it automatically. Additionally, you will need to add the GeoJson file created in Section 5 to the server file system.
    - Google Sheets Credentials (required for connecting the Python scripts to Google Sheets) have already been created for Banksia Gardens. If these credentials ever get corrupted or are not working refer to Section 6.
  + **Melbourne Airport Data**
    - Currently this worksheet is being automated by Banksia Gardens’ linux server by following the directions in Section 1.
      * The server is set to automatically refresh the current worksheet. If you would like to create a new worksheet, you must manually do so using Section 3.
    - This worksheet should be monitored frequently, and if there are problems with the data upload you can manually update the sheet by following Section 3.
    - Google Sheets Credentials (required for connecting the Python scripts to Google Sheets) have already been created for Banksia Gardens. If these credentials ever get corrupted or are not working refer to Section 6.
    - An OpenWeather API key was provided to Banksia Gardens (in the code) to extract the temperatures. If this key ever gets corrupted or needs to be replaced, create an account on openweathermap.org and request an API key. It will be sent to you via email.
  + **Heat Gun Data**
    - This worksheet is set up for manual management. See Section 4 for directions.
  + **Temperature Sensor Data**
    - All data uploaded to this worksheet is managed by Raspberry-Pi-Controlled temperature sensors created for Banksia Gardens. Directions on how to recreate them can be found in the “Constructing and Developing Heat Sensors” manual ([linked here](https://docs.google.com/document/d/1STVYR2q_rWUhKjV4XfgRBDcpeL7PxhaaMI647ZXKwHE/edit?usp=share_link)).
    - If there is a problem with this worksheet, you can view the error logs described in the “Constructing and Developing Heat Sensors” manual linked above
    - Additionally, there are currently seven worksheets (one for each sensor) in a separate Google Sheet file ([linked here](https://docs.google.com/spreadsheets/d/1Nk45Gvtw8gvpxw4DNpza-c6WmaWeUOcJXAfeR9I5KBQ/edit?usp=share_link)) that can be used to locate which sensors are not working properly
* Other Important Notes
  + If running the biodiversity and Melbourne Airport code on the linux server, it would be a good idea to periodically check the ErrorLogger.txt file (located in the same folder on the server). This file will record any errors occurring during the automation.
  + Do not update the biodiversity code more than once a day (It is scheduled to run at 12:00 am each day but this can be changed to a different time).
    - This can cause the script to delete and re-add too many data points which may result in an error
  + Do not change anything with the Google Sheet structure or Python code unless addressed properly through directions from the manual
    - This includes the title of the Google Sheet, Worksheet names, Python script names, etc.
  + Section 7 gives descriptions for all column names in each Worksheet
  + For specific operational questions feel free to contact the authors of this manual
    - Eric Schuman
      * Email: [eschuman20@gmail.com](mailto:eschuman20@gmail.com)
      * LinkedIn: <https://www.linkedin.com/in/eric-schuman-b73aa01b3/>
    - Lily Bromberger
      * Email: [lily.bromberger@gmail.com](mailto:lily.bromberger@gmail.com)
      * LinkedIn: <https://www.linkedin.com/in/lily-bromberger/>

# Section 1: Setting Up Automation Of iNaturalist and OpenWeather Data Collection and On A Linux Server

1. **Download Termius from this link:** [**https://termius.com/free-ssh-client-for-windows**](https://termius.com/free-ssh-client-for-windows)
2. **Connect to your Linux Server Through Termius**
3. **In Termius, double click on Hosts, and then on your server to open the server**
4. **Ensure that Python is downloaded in the Linux Environment**

Type “python3 --version” in the terminal to see if it is already installed. Use the command ‘apt-cache search python3’ to see available versions of Python for download. We used Python3.8 but any version 3.8 or newer should work.

**If it is not installed…**

Install it onto the Linux server using the following command in the terminal: “sudo apt-get install python3.X” where X is whichever version you wish to download. If you have problems with this, there are many guides on the internet on installing Python on linux servers

**If installed & wrong version…**

Use the “sudo apt-get install python3.8” to upgrade the version.

**You may also need to update the symbolic link:**

Use the command “sudo update-alternatives --install /usr/bin/python python /usr/bin/python3.8 1”. This now tells the system to use that version of Python.

Verify with python --version to ensure that you have done this correctly.

1. **Install Pip**

Use command “sudo apt install python3-pip”

\*Note: If pip is already installed, make sure it is the right version by using the command “pip –version”. To upgrade pip, type in the command “python3 -m pip install –upgrade pip”

1. **Install required libraries**

In the terminal, install each of these libraries by copy and pasting these lines and clicking enter.

For iNaturalist automation:

**pip3 install pyinaturalist**

**pip3 install pyinaturalist\_convert**

**pip3 install os**

**pip3 install shapely**

**pip3 install gspread**

**pip3 install gspread\_dataframe**

**pip3 install oauth2client.service\_account**

**pip3 install datetime**

**pip3 install json**

**pip3 install pandas**

For OpenWeather automation:

**pip3 install pyowm**

pip3 install gspread

pip3 install gspread\_dataframe

pip3 install oauth2client.service\_account

pip3 install datetime

pip3 install json

pip3 install pandas

\*Note: if this doesn’t work try command “python3 -m pip install XXXXX”

1. **Check download**

Type “python3” in the terminal. This will open a Python environment. Now if you type “import XXXX” where XXXX is the library (such as pyowm), and you hit enter, a new line will appear with no text (it will look like this: >>>). This means that the library has been installed correctly. If this does not occur try re-installing.

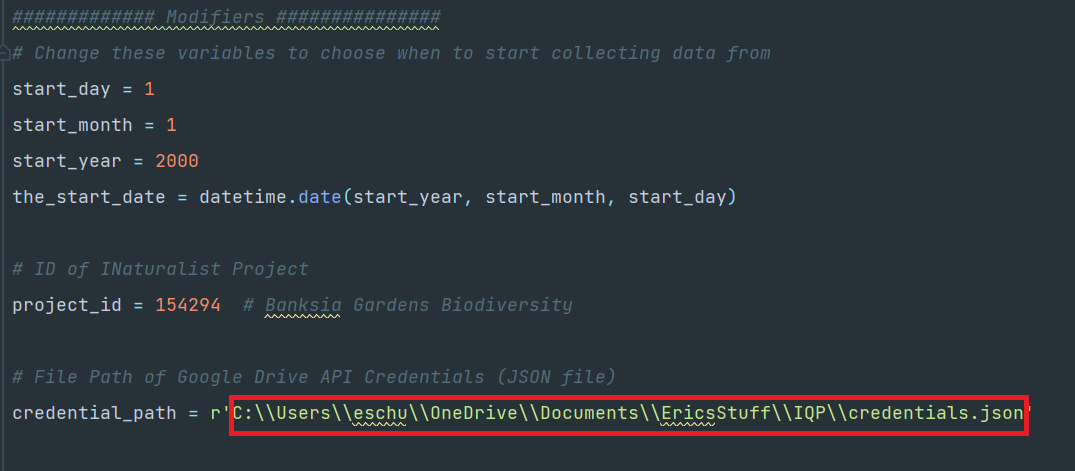
To exit the Python editor type exit() and hit enter

1. **Install a Python editor**:

You can install the latest version of pycharm at the following link:<https://www.jetbrains.com/pycharm/download/#section=windows>. Ensure you download the version compatible with your computer. The community version of this software is okay. Any alternative Python IDE should also work.

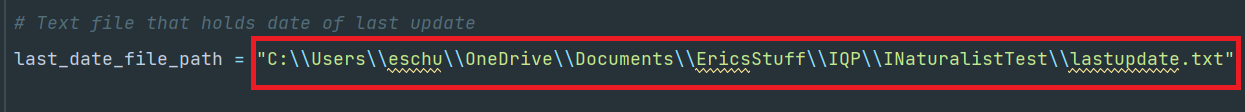
1. **Open the code in the file biodiversity.py (found in the database manual folder) in your Python editor. \*Note: make sure you choose the version that includes error logging**
2. **Update Python code with the JSON credentials file path to match the new path to the file. (found in the database manual folder) \*Note: if you need to create a new credentials file because the old one was compromised or for some other reason, see Section 6.**

The file credentials.json is included in the database manual folder. Right click the file and select “copy as path.” Paste the path in the credentials\_path variable. \*Note: Make sure you do not delete the “r” coming before the path string and depending on the editor you may need to have two backslashes (\\) between each subdirectory instead of one.

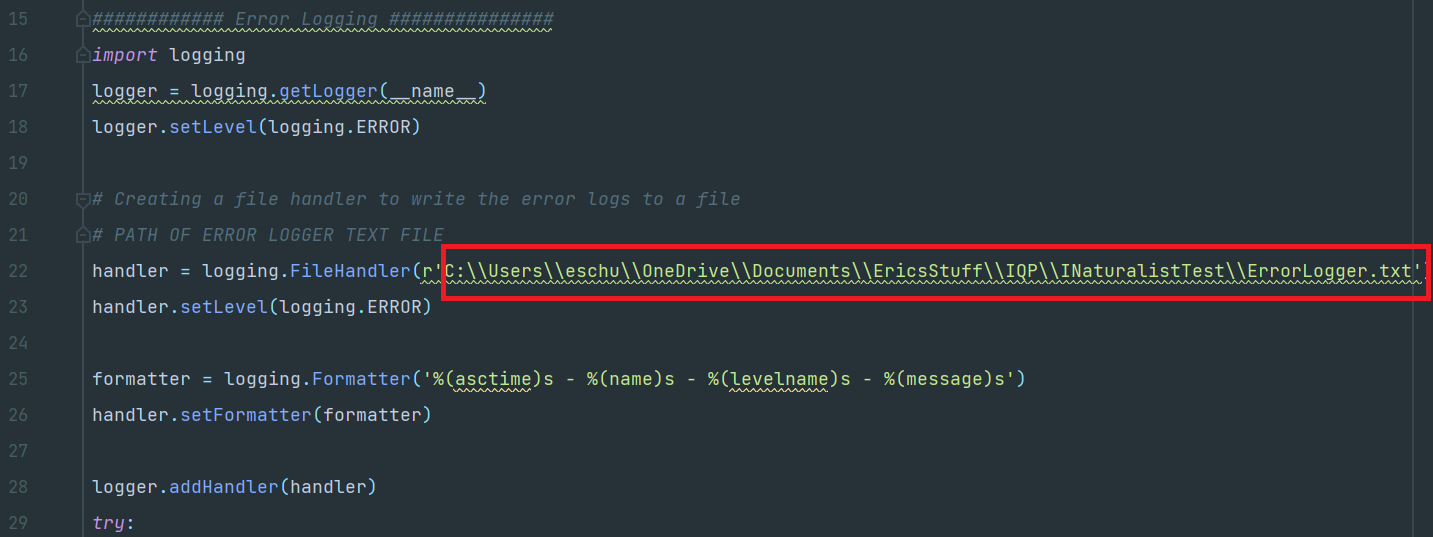
****

1. **Locate the lastupdate.txt and ErrorLogger.txt files (included in the database manual folder) and fix their file paths.**

Open the last update text file and make sure the date is updated to the date you want to begin updating from iNaturalist (yyyy-mm-dd). This date will automatically update when automation is set up. Follow the same process detailed in Step 10 of copying the path to the text file into the Python file.

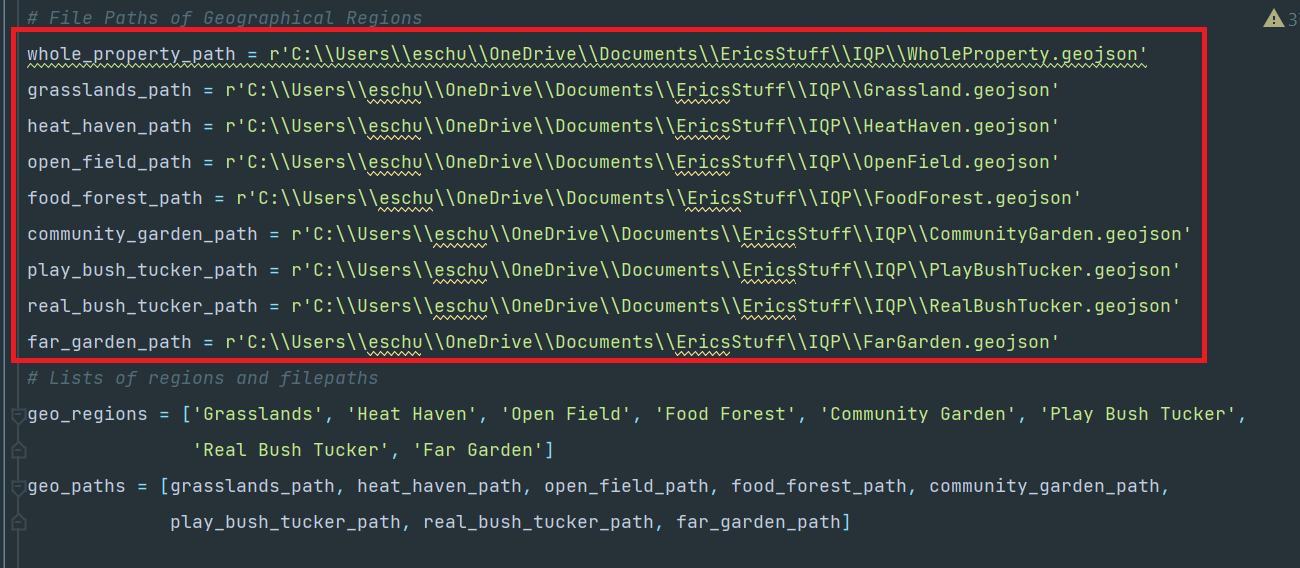


The error logger text file is used to record any errors that may occur while the script is being automated. The database manager can refer to it whenever there are issues with the database. Follow the same process detailed in Step 10 of copying the path to the text file in the Python file (located above the Modifiers Section of the code).



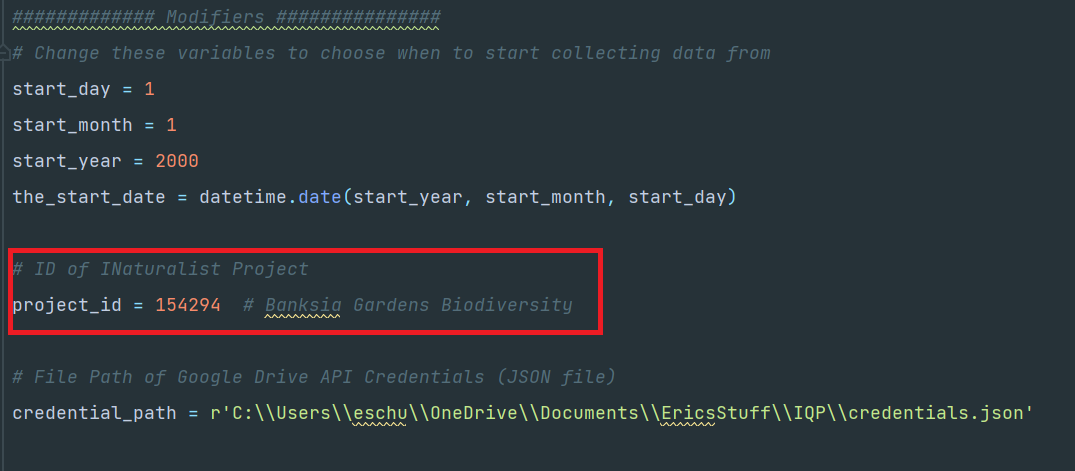
1. **Find the GeoJSON files (located in the database manual folder) detailing the coordinate boundaries of different locations to be tracked. Update these as well.**

Follow the same process detailed in Step 10 of copying the path into the Python file. (\*Note: If you would like to alter, delete, or add a new location, see Section 5. This will likely need to happen if there are expansions or creations of new regions)

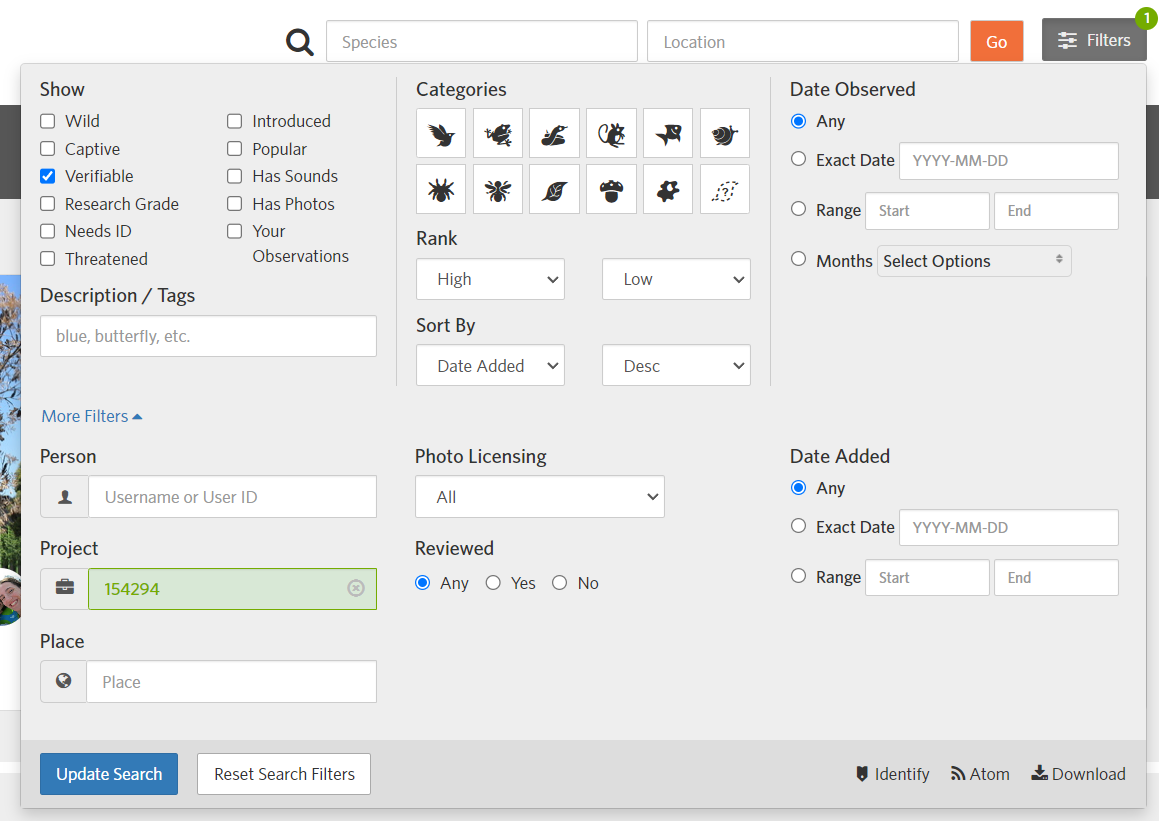
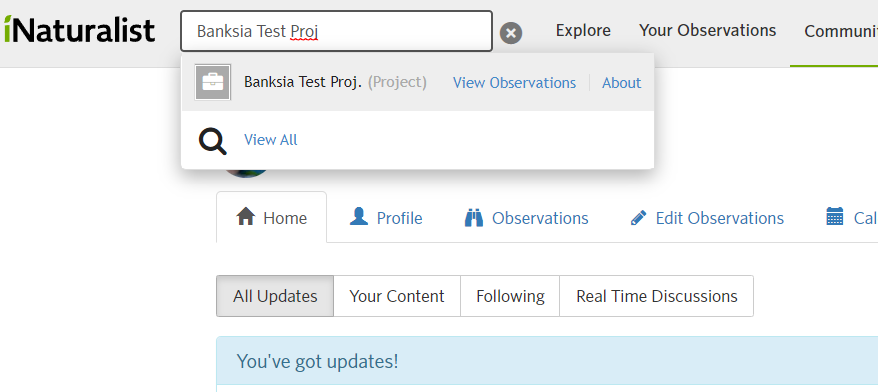


1. **Select the project ID you are using**

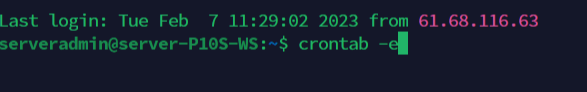
Ensure that you put the right project ID in the modifiers section of the code

**

\*Note: to find the project ID go into iNaturalist and search the name of the project. Select view observations. Click on the filter icon.



1. **Check the rest of the variables listed in the “Modifiers” section of the code** 
   1. The google sheet name and worksheet name reflect the sheet and worksheet that the data will be copied into. If you change the worksheet name to a new name, it will create a new worksheet in the google sheet database.
   2. The datatable column structure (order of columns in datatable) as well as specific Google Sheet ranges are also listed in the modifiers. ONLY change these variables if the Google Sheet structure is being altered by adding or deleting columns. This will cause the columns to shift which may break the code unless the positions of these specific columns in the datatable are also updated.
   3. Instructions around changing the lists of geo regions and path variables can be found in Section 5.
   4. Don’t modify the section of code at the end of the modifiers sections unless running the code manually (not automated).
2. **Open the code in the file temperature.py (found in the database manual folder) into your Python editor. \*Note: make sure you choose the version that includes error logging**
3. **Fix the filepath of the ErrorLogger.txt file by following the same process used in Step 11.**
4. **Repeat Step 10 to edit the JSON credential file path for uploading data to google sheets in temperature.py**
5. **Check the rest of the variables listed in the “Modifiers” section of the code.** 
   1. The google sheet name and worksheet name reflect the sheet and worksheet that the data will be copied into. If you change the worksheet name to a new name, it will create a new worksheet in the google sheet database.
   2. The latitude and longitude coordinates can be adjusted to change the location that the weather data is being collected from.
   3. The OpenWeather API Key should not need to be altered (unless you do not have a key yet). If you do not have a key or it needs to be replaced, create an account on openweathermap.org and request an API key. It will be sent to you via email.
6. **Click on SFTP in the sidebar and create a new folder in the linux server**
7. **Locate the database manual folder on your local computer and copy all of the files over into the new folder created in Step 19**
8. **Go to the Linux terminal and type “crontab -e” and hit enter**



1. **Schedule the running of a script (do this for both biodiversity.py and temperature.py)**

Type: \* \* \* \* \* /path/to/python3 /path/to/script/to/run

Where the five \*’s represent minute (0-59), hour (0-23), day of month (1-31), month (1-12), day of week (0-6, Sunday = 0). You can also leave an \* to represent all or any for each field based on how often you want the script to run. The path to python3 can be found by typing in “which python3” in the terminal. The path to the script you are running depends on which script (biodiversity.py or temperature.py) and where it is located. You can also add a @reboot before the command to ensure that every time the server reboots, the file is rerun.

**Here is an example:**

**#Temperature Every Hour rerun file**

**0 \* \* \* \* /usr/bin/python3 /home/serveradmin/Biodiversity\_Automation/temperature.py**

**#Biodiversity Every Hour rerun file**

**0 \* \* \* \* /usr/bin/python3 /home/serveradmin/Biodiversity\_Automation/biodiversity.py**

**#Biodiversity on reboot**

**@reboot /usr/bin/python3 /home/serveradmin/Biodiversity\_Automation/biodiversity.py**

1. **Save and exit**

Press ctrl + X to prompt exit

then Y to save

then Enter to close the crontab window.

1. **Check if it worked**

Back in the regular terminal, type : crontab -l to see it display the most recent crontab opened. If it is NOT updated to what you just edited, repeat steps 21-23.

# Section 2: Setting Up Manual Data Collection from iNaturalist

1. **Install Python**

Install Python’s latest version at the following link:<https://www.python.org/downloads/>. Ensure you download the version compatible with your computer.

1. **Install a Python editor and open the biodiversity.py file (found in the database manual folder). \*Note: choose the version without error logging**

Follow Step 8 in Section 1 to install the Python editor.

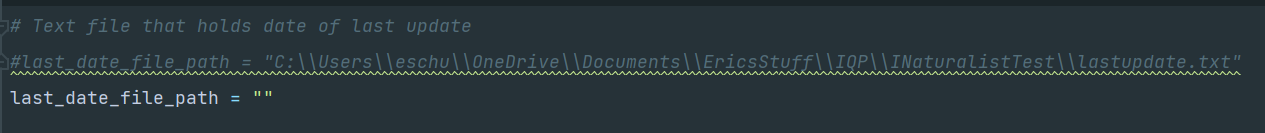
1. **Check pip installation in the editor**

Within your editor, go to the terminal and type “pip --version”. If it is not installed, this will tell you. If pip is not installed, look up how to install it in your specific editor.

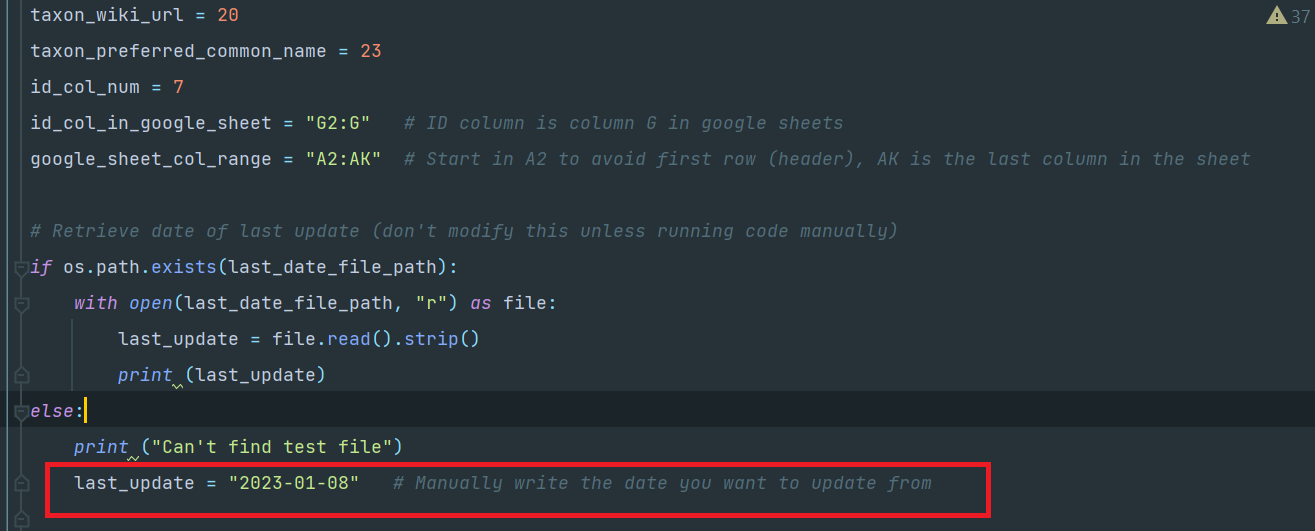
1. **Install libraries needed from Section 1 Step 6**

In the terminal, install each of the libraries listed in Section 1 Step 6 under “For iNaturalist Automation.” Use the command “pip install xxxx” instead of “pip3 install xxxx”

1. **Complete Steps 10 and 12-14 from Section 1**
2. **Running functions manually**
3. First change the variable last\_date\_file\_path in the modifiers section to an empty string (“”). \*Note: recommended to comment out the variable and make a new one with the empty string so that the file path is still saved for future use. See below:



1. In the bottom of the modifiers section, change the last\_update variable to the date you wish to be updating or collecting data from.



1. In the “Collecting Data” section of the code (at the bottom), there are commands you can manually run.

* To create a new worksheet, uncomment the code under “Create Initial Sheet” and run the file. Make sure the code under “Refresh” and “Automatic Refresh” is commented out
* To manually update a worksheet, uncomment the code under “Refresh” and run the file. Make sure the code under “Create Initial Sheet” and “Automatic Refresh” is commented out
* Once completed, if you manually update the sheet every time it would be a good idea to change the last\_update variable (yyyy-mm-dd) mentioned in Step 6.b to reflect the current day you are updating from so the code is already set for the next update.

# Section 3: Setting Up Manual Data Collection from OpenWeather

**1.** **Install Python**

Install Python’s latest version at the following link:<https://www.python.org/downloads/>. Ensure you download the version compatible with your computer.

**2. Install a Python editor and open the temperature.py file (found in the database manual folder). \*Note: choose the version without error logging**

Follow Step 8 in Section 1 to install the Python editor.

**3.** **Check pip installation in the editor**

Within your editor, go to the terminal and type “pip --version”. If it is not installed, this will tell you. If pip is not installed, look up how to install it in your specific editor.

**4. Install libraries needed from Section 1 Step 6**

In the terminal, install each of the libraries listed in Section 1 Step 6 under “For Temperature Automation.” Use the command “pip install xxxx” instead of “pip3 install xxxx”

**5.** **Complete Steps 17-18 in Section 1**

**6. Running functions manually**

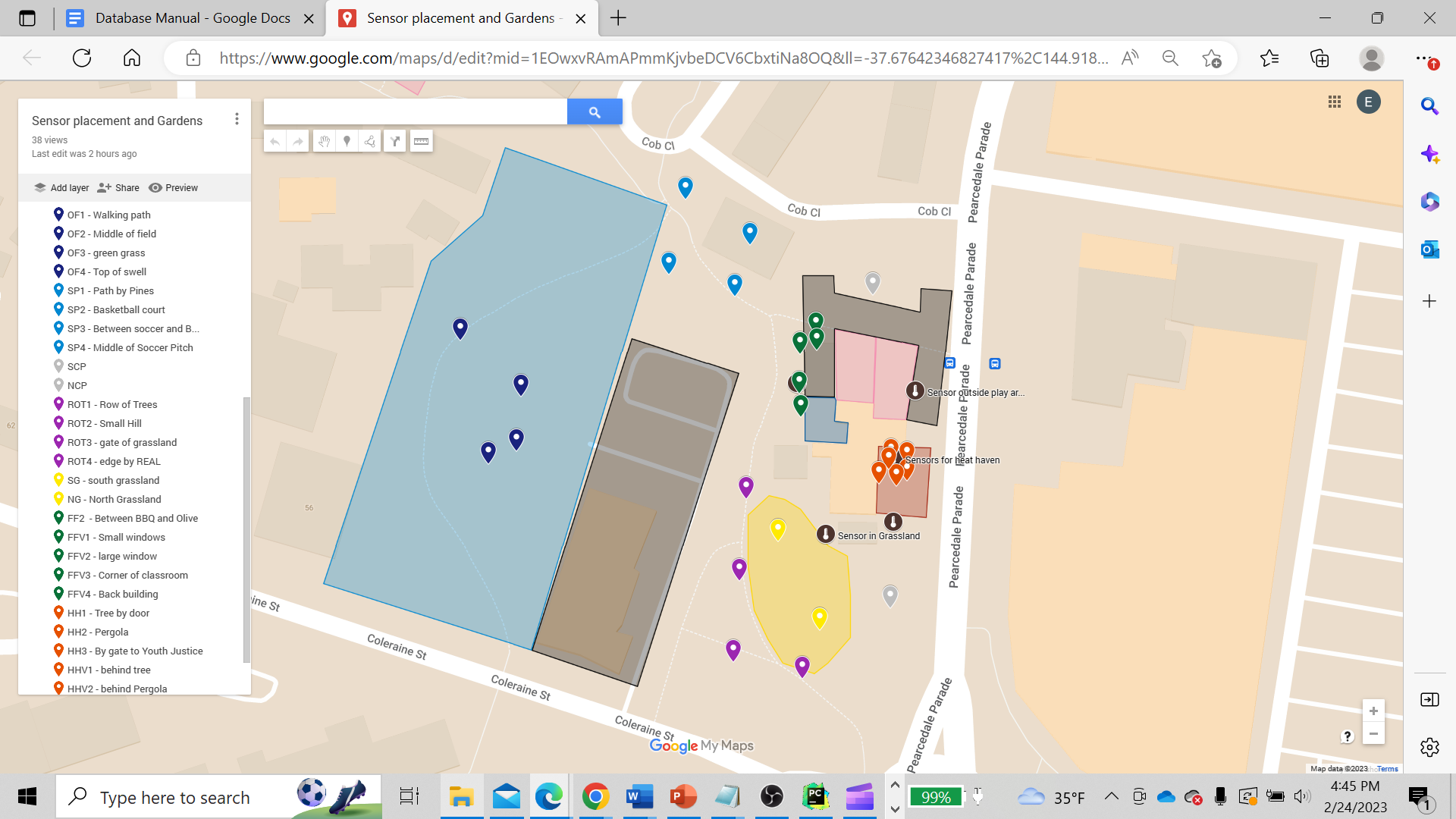
Follow Step 6.c from Section 2. (You can ignore the part about changing the last\_update variable)

# Section 4: Managing Heat Gun Data

\*Note: This entire table in the database in managed manually

1. **To record a new entry to the database, take heat recordings at each of the locations outlined in the figure and link below.**

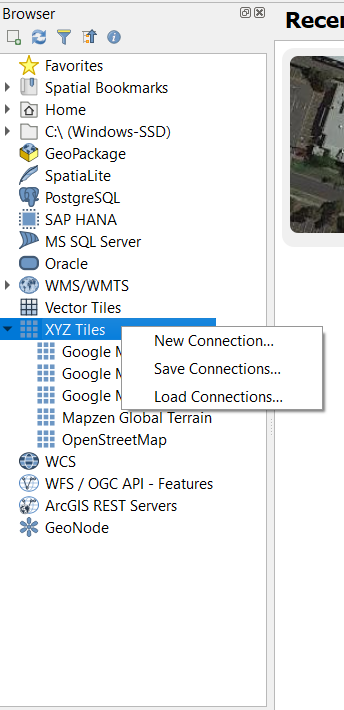
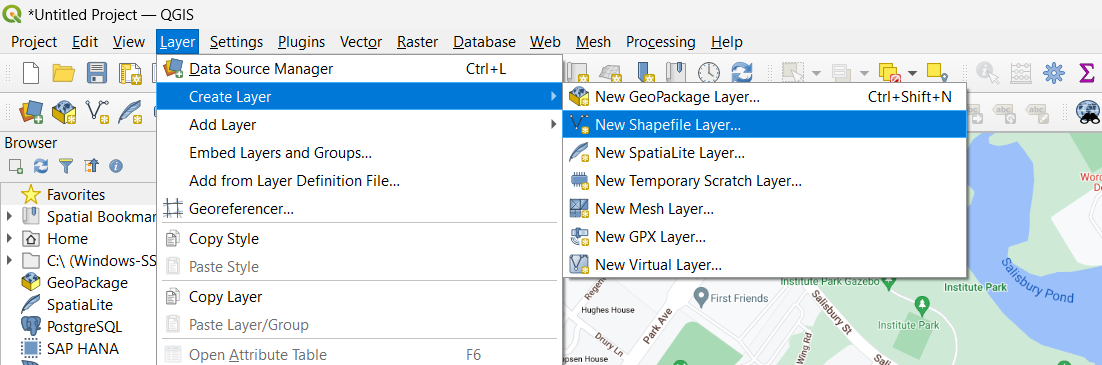
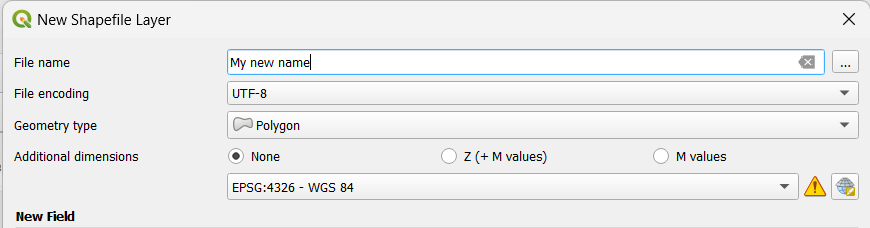
[**Click this link for an interactive view**](https://www.google.com/maps/d/edit?mid=1EOwxvRAmAPmmKjvbeDCV6CbxtiNa8OQ&ll=-37.67643854087057%2C144.91806386951234&z=18)

****

1. To make an observation, take the heat gun and point it towards the ground
2. Press the button on the handle of the heat gun
3. Record the temperature that is displayed on the screen
4. **After taking recordings at each location, go to the worksheet labeled “Heat Gun Data” in the Biodiversity and Temperature Database and manually type in all recordings in their respective columns.**

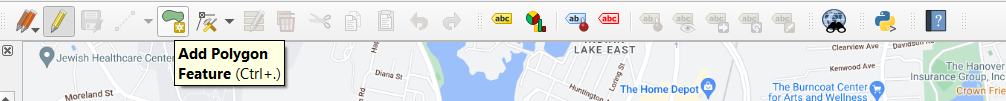
# Section 5: Adding GeoJSON Files to Reflect the Coordinate Boundaries of a Specific Location

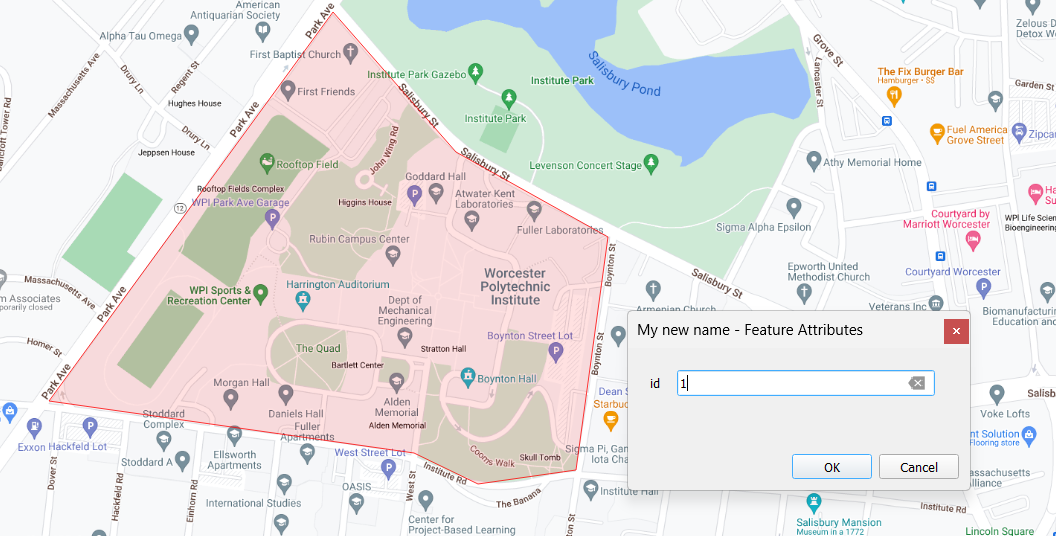
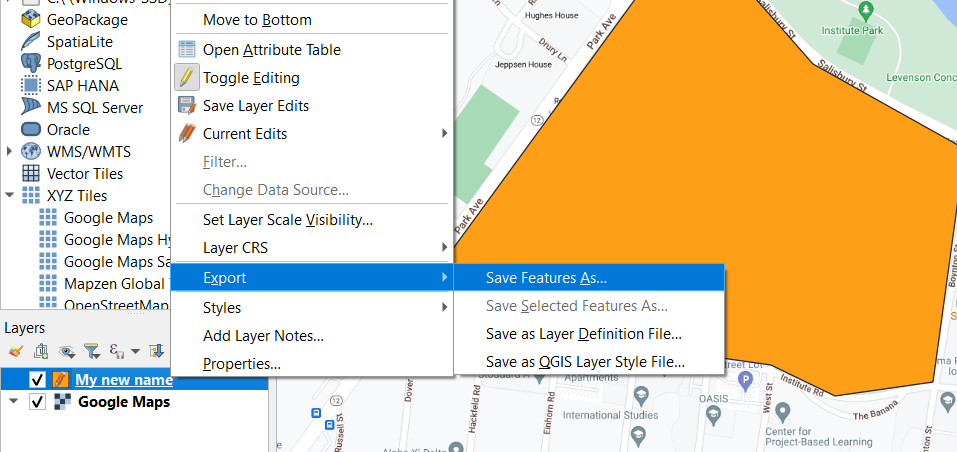
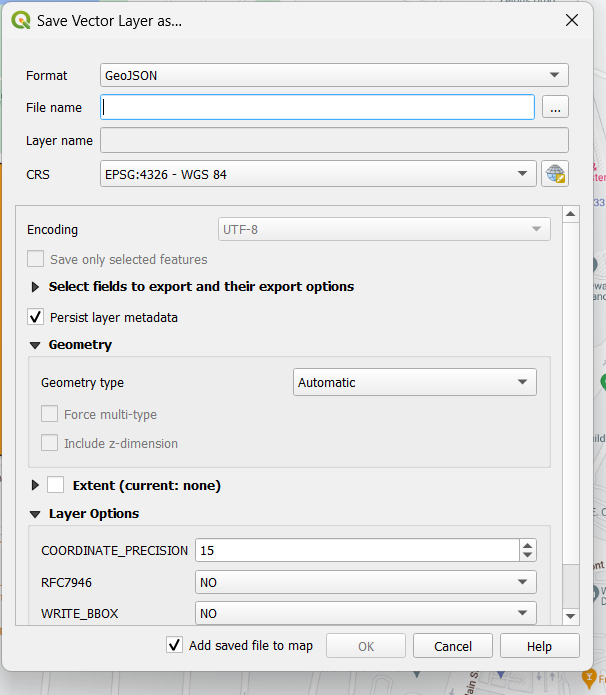
Creating a new GeoJSON File:

1. **Download and Install QGIS from the internet.**
2. **Open the QGIS application**
3. **On the left side of the application, right click XYZ tiles and select new connection**
4. **Name the connection anything you want and then add one of the following in the URL section:** [**Link**](https://hatarilabs.com/ih-en/how-to-add-a-google-map-in-qgis-3-tutorial)
   1. **Roadmap** - http://mt0.google.com/vt/lyrs=m&hl=en&x={x}&y={y}&z={z}
   2. **Terrain -** http://mt0.google.com/vt/lyrs=p&hl=en&x={x}&y={y}&z={z}
   3. **Altered roadmap -** http://mt0.google.com/vt/lyrs=r&hl=en&x={x}&y={y}&z={z}
   4. **Satellite only -** http://www.google.cn/maps/vt?lyrs=s@189&gl=cn&x={x}&y={y}&z={z}
   5. **Terrain only -** http://mt0.google.com/vt/lyrs=t&hl=en&x={x}&y={y}&z={z}
   6. **Hybrid -** [http://mt0.google.com/vt/lyrs=y&hl=en&x={x}&y={y}&z={z](http://mt0.google.com/vt/lyrs=y&hl=en&x=%7Bx%7D&y=%7By%7D&z=%7Bz)}
5. **If the Map view doesn’t open up, click on the created XYZ view.**
6. **Zoom into your region.**
7. **Create a new shapefile layer.**
8. **Create a new shapefile. Add the name of choice and then change the file encoding to UTF-8. Change the Geometry type to Polygon. Select “ok”**

You new shape will appear on the left below the browser

1. **Along the 3rd row from the top, Select turn toggle on (pencil) and then Create new polygon**

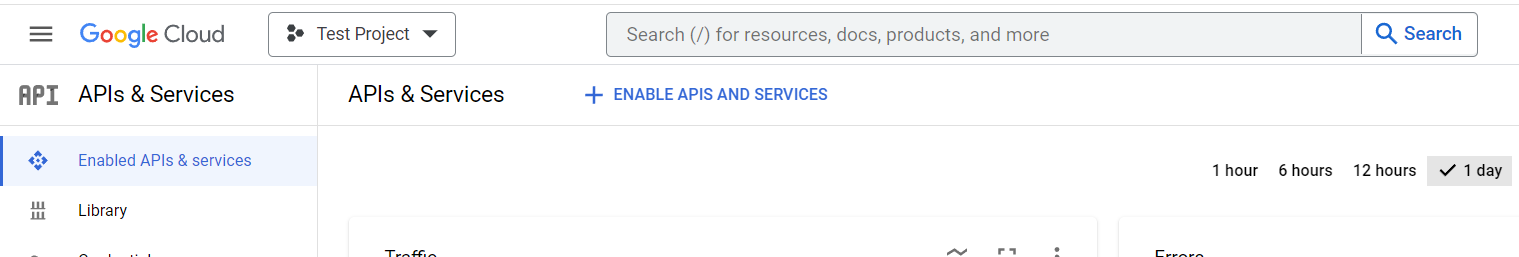
****

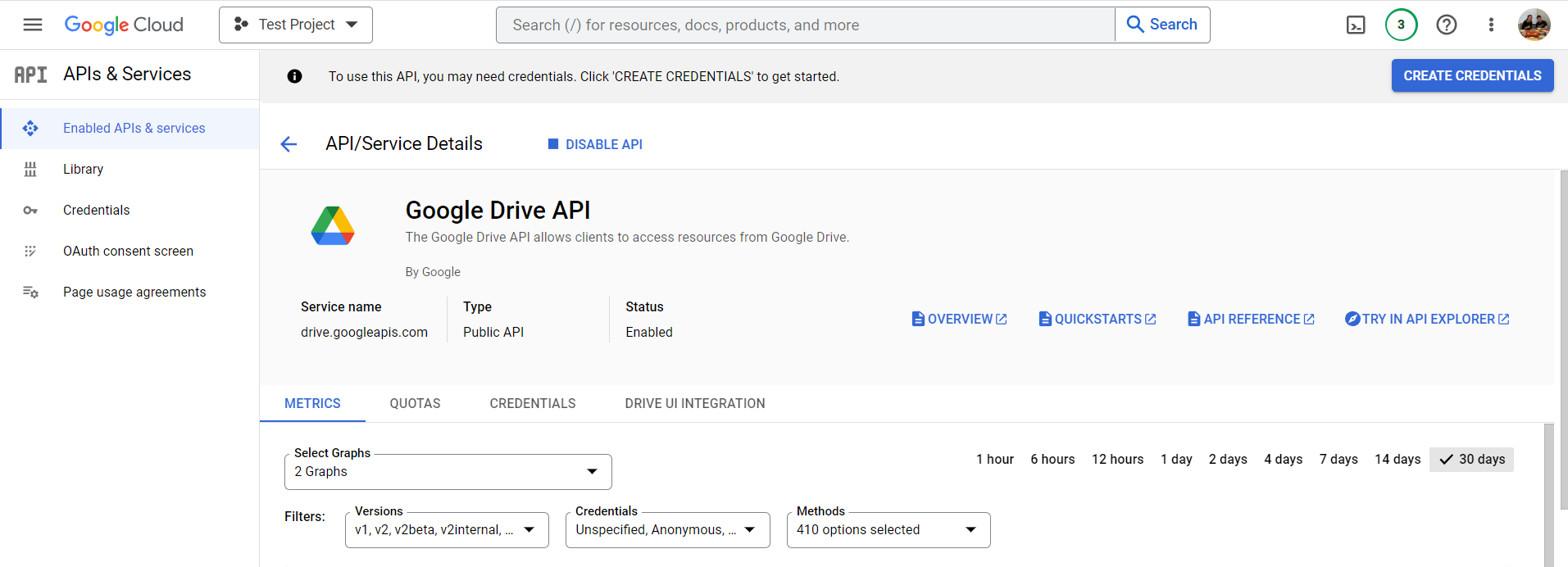
1. **Create your shape by left clicking the desired area. When you are satisfied, you can left click which will close the drawer. Then you type in a unique ID number for the shape.**
2. **To export this new shape right click on the shape name, then hover over export and then select “save features as…”**
3. **Select the GeoJSON format of the shape. In file name, select browse and name it according to the file location. Select “OK”.**
4. **Edit the biodiversity.py file to add, alter, or delete a new coordinate bound location (GeoJSON file). \*Note: If you are using the linux server to automatically update the biodiversity data, make sure you replace the old biodiversity.py file with the updated one in the project folder on the linux server (Do the same with any new or updated GeoJSON files)** 
   1. If altering a coordinate bound location currently being used in the biodiversity.py file, just replace the old GeoJSON file path for that location with the new GeoJSON file path.
   2. If adding a new coordinate bound location, create a new variable to hold the path of the GeoJSON file. Then add that variable to the end of the geo\_paths array. Additionally, add the name of that region to the geo\_regions array.
   3. If deleting a coordinate bound location, make sure you delete the variable holding the path to the GeoJSON file as well as the location name from the geo\_regions array and path from the geo\_paths array.

\*Note: The whole\_property\_path does not need to be included in the lists of regions and list of file paths. This is because it likely includes many or all of the other regions.

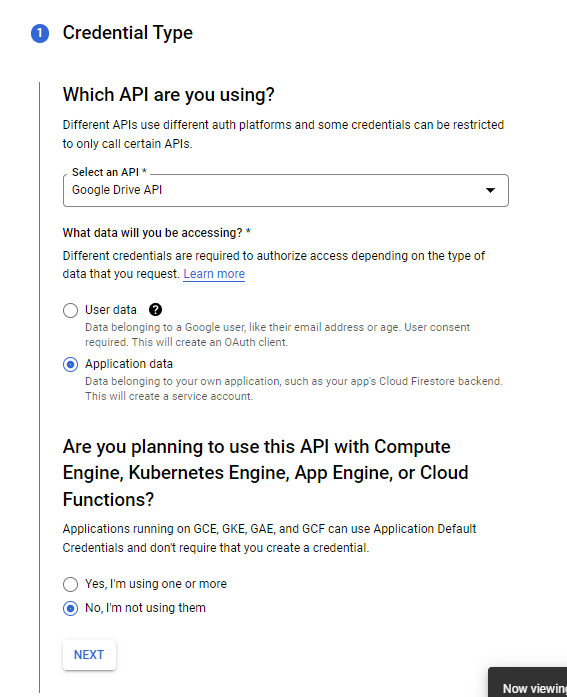
# Section 6: Creating New Google Sheets Credentials

\*Note: The google sheet credentials currently used in the files (and saved in the database manual folder) should work. You should only need to create new ones if the old credentials were lost or compromised.

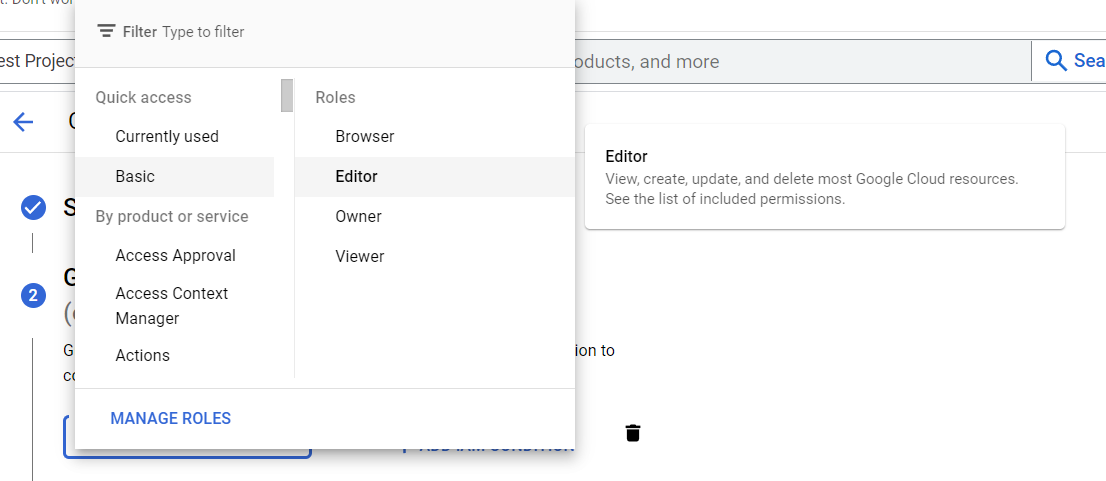
1. **Click on** [**this link**](https://console.developers.google.com/cloud-resource-manager) **to create a project in the Google Cloud Console.**
2. **Go to the project dashboard and hover over “APIS AND SERVICES”. Click “Enabled APIs & Services.”**
3. **Click +ENABLE APIS AND SERVICES **
4. **Search for and enable Google Drive API**
5. **Click on the blue “Create Credentials” box in the top right corner**



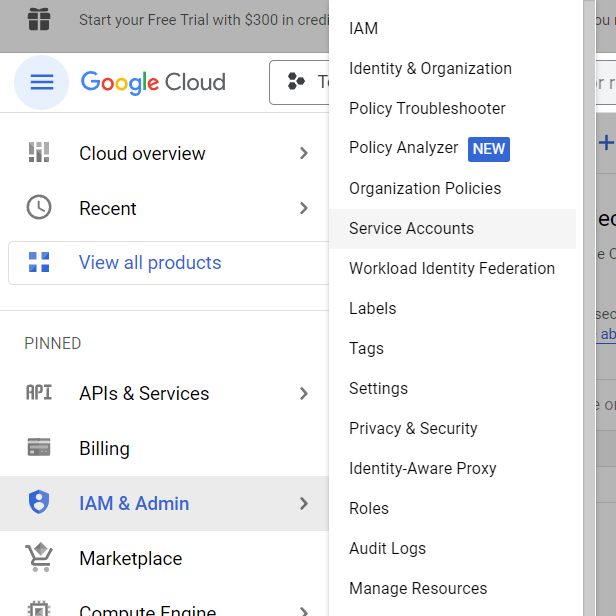
1. **Fill out the credentials like below**



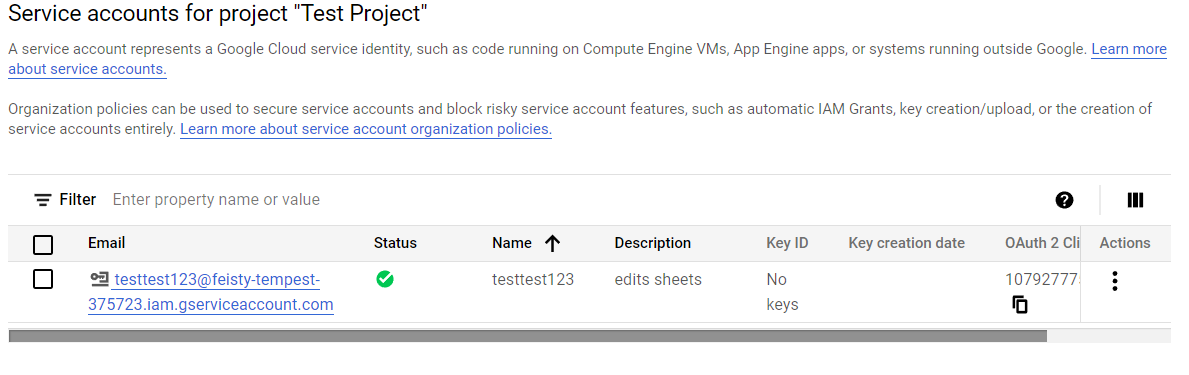
1. **Select next and you will be taken to a “Create Service Account” page**
2. **Fill in a unique name and select “Create And Continue”**
3. **Fill in role as a basic editor and hit continue, then hit done**



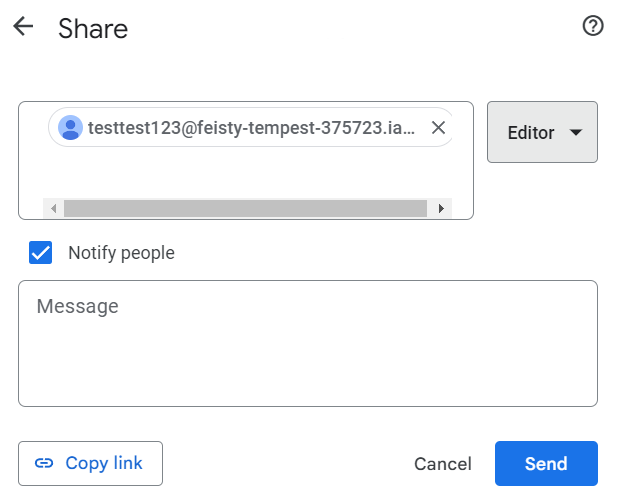
1. **Repeat steps 2-4 for the Google Sheets API**
2. **On the left, hover over“IAM & Admin”, then “Service Accounts”**



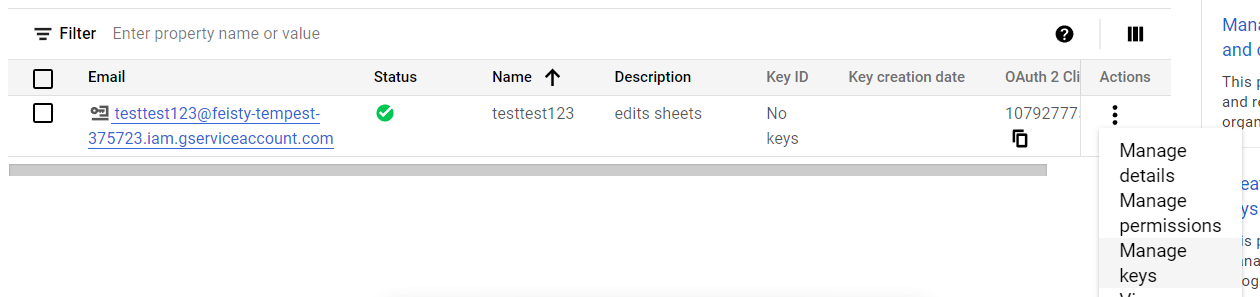
1. **Copy Email**

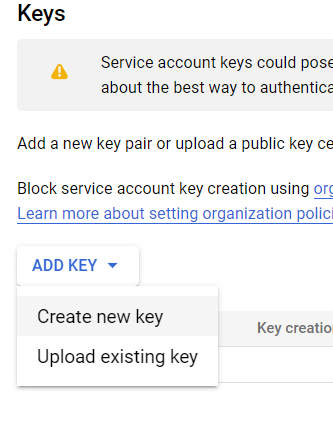


1. **Go to the Google Sheet and share the doc with the email. Allow the copied email to edit the document**

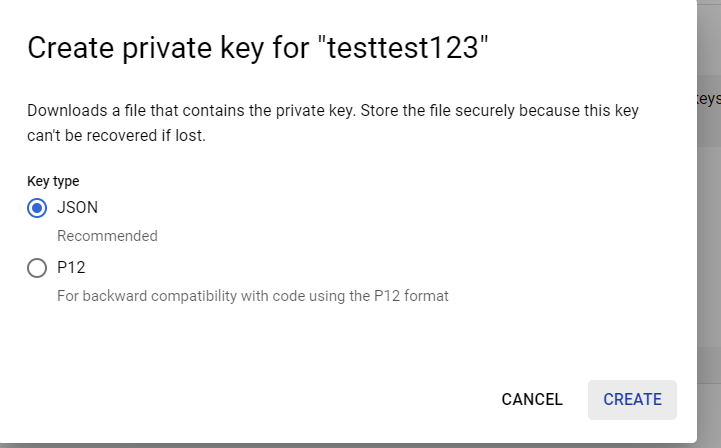


1. **Go back to the “Service Account” and select the 3 vertical dots under actions. Select “Manage Keys” >“ADD KEY” > “Create new Key”**





1. **Select the Key type as JSON and hit “CREATE”**



1. **Save the downloaded JSON file under the desired file path. \*Note: If replacing old credentials make sure to update the Python code with the JSON credentials file path to match the new path to the file.**

# Section 7: Database Column Descriptions

## 1. Biodiversity Data

| **Column** | **Description** |
| --- | --- |
| quality\_grade | research if the observation has been verified, need\_id if there is no consensus for the identification, casual if there is missing data or it is not wildlife |
| observed\_on\_details\_day | The date of the observation |
| observed\_on\_details\_month | The month of the observation |
| observed\_on\_details.hour | The hour of the observation |
| observed\_on\_details.year | The year of the observation |
| observed\_on\_details.day | The day of the observation |
| id | The id of the observation |
| identifications\_most\_agree | True if most of the species identifiers agree on what the species is |
| species\_guess | What the observer guessed the species was |
| identifications\_most\_disagree | True if most of the species identifiers disagree on what the species is |
| reviewed\_by | The id’s of the people that have reviewed/verified what the species is |
| description | Any notes left by the observer about the observation |
| updated\_at | Date and time that the observation was last updated (last update is usually if there is a new reviewer) |
| taxon.endemic | True if the species can only be found in this general area |
| taxon.threatened | True if the species is threatened |
| taxon.introduced | True if the species was introduced to the area |
| taxon.native | True if the species is native to the area |
| taxon.name | Name the organism is classified under by taxonomists |
| taxon.rank | The rank of the species in taxonomic hierarchy (how specific the classification is) |
| taxon.extinct | Whether or not the species is currently extinct |
| taxon.id | Id given by taxonomists |
| taxon.wikipedia\_url | Wikipedia url for the species |
| taxon.default\_photo.medium\_url | Photo given by iNaturalist for the species |
| taxon.iconic\_taxon\_name | The general type of species |
| taxon.preferred\_common\_name | The common name of the species |
| num\_identification\_agreements | The number of identification agreements on what the species is |
| comments | Any comments that other users have made about the observation |
| uri | The link to the observation on iNaturalist |
| geojson.coordinates | The coordinates of where the observation was made |
| user.login | The username of the observer |
| photo\_url | The photo taken by the observer |
| number\_of\_reviews | The number of reviews/verifications made |
| australian\_season | The season the observation was made in |
| aboriginal\_season | The aboriginal season the observation was made in |
| time\_of\_day | The relative time of day the observation was made (morning, afternoon, evening, night) |
| latitude | The latitude coordinate the observation was made at |
| longitude | The longitude coordinate the observation was made at |
| region | The region in Banksia Gardens the observation was made at |
| on\_property? | True, if the observation was made on Banksia Gardens property |

## 2. Melbourne Airport Data

| **Column** | **Description** |
| --- | --- |
| date | The date of the recorded temp |
| hour | The hour of the recorded temp |
| temp | The temperature in degrees celsius |
| humidity | The humidity percentage (out of 100) |

## 

## 3. Heat Gun Data

| **Column** | **Description** |
| --- | --- |
| Date | The date of the recorded temp |
| Time | The time of the recorded temp |
| Melbourne Airport | The temperature from Melbourne Airport |
| HH1 - Heat Haven 1 | Under the tree by the front door |
| HH2 - Heat Haven 2 | Under the Pergola |
| HH3 - Heat Haven 3 | Ground outside of Youth Alliance Hub |
| HHV1 - Heat Haven Vertical 1 (pointed at a wall) | Behind Tree |
| HHV2 - Heat Haven Vertical 2 (pointed at a wall) | Behind Pergola |
| HHV3 - Heat Haven Vertical 3 (pointed at a wall) | Youth Justice Wall |
| NCP1 - North Car Park | Center of the Car Park by the Community Garden |
| FF1 - Food Forest 1 | In front of small windows |
| FF2 - Food Forest 2 | In front of big window |
| FF3 - Food Forest 3 | Corner of classroom |
| FF4 - Food Forest 1 | Between BBQ and Olive Tree |
| FFV1 - Food Forest Vertical 1 (wall) | Classroom wall, small windows |
| FFV2- Food Forest Vertical 2 (wall) | Classroom wall, big window |
| FFV3 - Food Forest Vertical 3 (wall) | Corner of classroom |
| SP1 - Soccer Pitch 1 | Path by pines |
| SP2 - Soccer Pitch 2 | Basketball court |
| SP3 - Soccer Pitch 3 | Between Soccer Pitch and Basketball Court |
| SP4 - Soccer Pitch 4 | Middle of Soccer Pitch |
| OF1 - Open Field 1 | Walking path |
| OF2 - Open Field 2 | Middle of field |
| OF3 - Open Field 3 | Green Grass |
| OF4 - Open Field 4 | Top of Swell |
| NG - North Grassland | In field, by Project Real |
| SG - South Grassland | In field, by parking lot |
| RT1 - Row of Trees 1 | By Car Park |
| RT2 - Row of Trees 2 | Small Hill |
| RT3 - Row of Trees 3 | Gate to Grassland |
| RT4 - Row of Trees 4 | Edge of Project Real |
| SCP1 - South Car Park | Center of the Parking Lot by the grassland |

## 

## 4. Temperature Sensor Data

| **Column** | **Description** |
| --- | --- |
| Time (UTC) | The time in Universal Coordinate Time |
| Temperature (°C) | The temperature in degrees celsius |

## 

# Citations

Images:

Google. (2021). Google Developers text logo [Google Developers logo].

<https://developers.google.com/>.

iNaturalist (California Academy of Sciences & National Geographic). INaturalist text logo [The logo of iNaturalist - citizen science project and website]. <https://www.inaturalist.org/>

Openweather. OpenWeather-Master-Logo RGB [Openweather logo]. <https://openweathermap.org/>

Python. Python-logo-notext. [Python logo]. [www.python.org](http://www.python.org).

QGIS software. (2020). QGIS logo minimal [Latest logo in 2020]. <https://www.qgis.org/en/site/getinvolved/styleguide.html>

Data Sources:

Banksia Gardens Biodiversity [Data set]. iNaturalist. [Banksia Gardens Biodiversity · iNaturalist](https://inaturalist.ala.org.au/projects/banksia-gardens-biodiversity)

*OpenWeather*. (n.d.). Weather Forecasts, Nowcasts and History in a Fast and Elegant Way. Retrieved February 17, 2023, from <https://openweathermap.org/>