Caroline Baumann - Jet Propulsion Laboratory, California Institute of Technology

SHARPENING CODE - MAC

ECOSTRESS TUTORIALS

This tutorial will show you how to use the ECOSTRESS Sharpening code on MacOS.

Table of Contents

Prerequisites

Before you start this tutorial, make sure you have an Earthdata Login, Visual Studio Code downloaded and set up, and a Python Environment to work with. You also need ECOSTRESS LST data downloaded on your computer for this tutorial. If you do not have ECOSTRESS data downloaded, you can either download it before starting this tutorial or you can follow the Sharpening API Code tutorial that uses and API to download both ECOSTRESS and Sentinel-2 data. If you need help setting any of these up or downloading ECOSTRESS data, please visit https://ecostress.jpl.nasa.gov/tutorials where you can follow along with the provided tutorials before proceeding with this one. This tutorial will walk you through an example of sharpening images of Dodger Stadium in summer of 2024, but you can follow along with whatever area and time of interest you want.

What is Sharpening and what is an API?

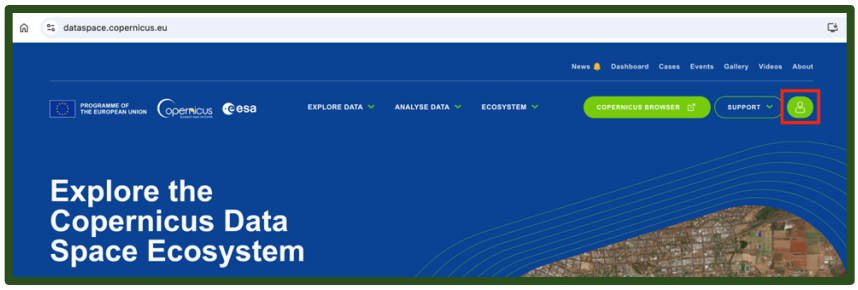
In remote sensing, image sharpening refers to enhancing the spatial resolution of satellite images in order to make them look more detailed. We use high resolution images to train a machine-learning model which is then used to sharpen low resolution images. In this code, 70-meter resolution ECOSTRESS data will be sharpened with 20-meter Sentinel-2 data. This tool uses an API (Application Programming Interface) to download Sentinel-2 data that will be used to sharpen ECOSTRESS LST images that you already have downloaded on your computer.

What is Copernicus Data Space?

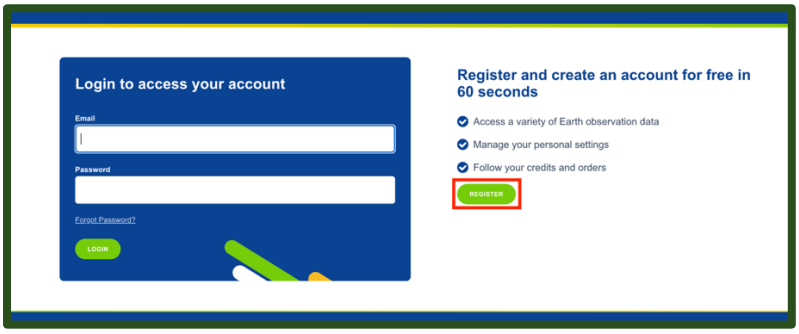
Copernicus Data Space is a European Space Agency platform that provides open data from the Copernicus Earth observation satellites, including Sentinel-2 data. We need a Copernicus Data Space login in order to create and use the Sentinel Hub API. An API, or Application Programming Interface, is a tool that allows your code to request and retrieve data from a server or database automatically.

CREATING A COPERNICUS LOGIN

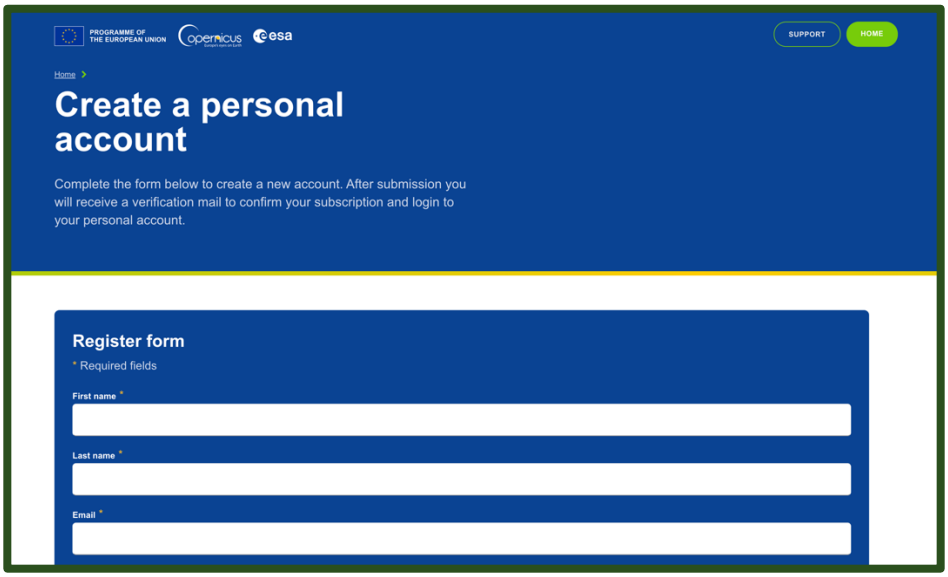
1. Start by going to https://dataspace.copernicus.eu/ or by searching the web for **Copernicus Data Space** and clicking on the first link. On the website, click the **green profile icon**.



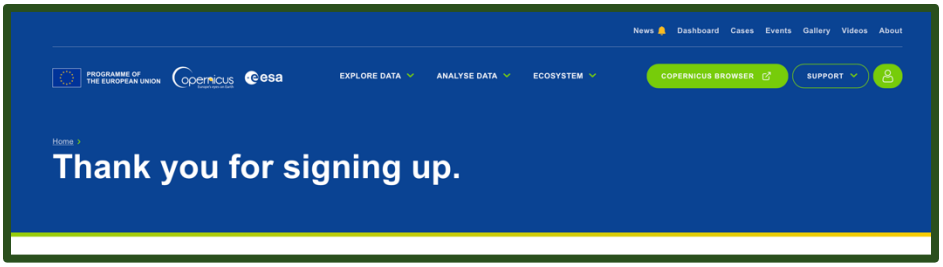
2. This will take you to the login page. If you already have an account, you can log in and move on to the next section of the tutorial. If you do not have an account, click the green register button.



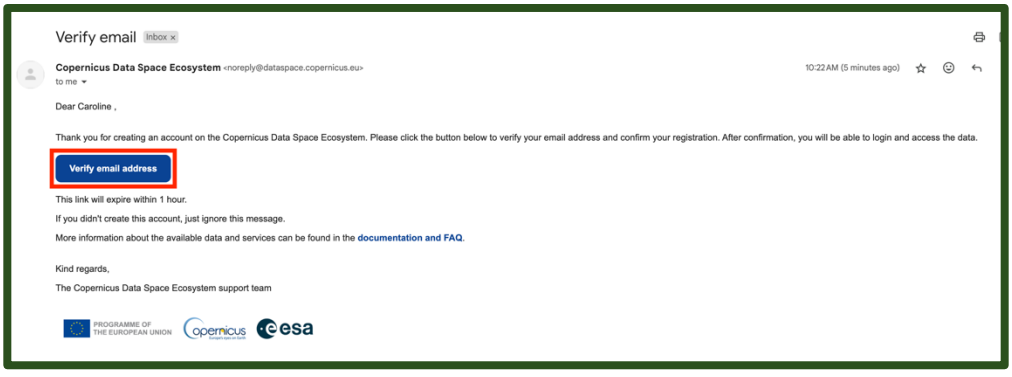
3. Fill out all of the required fields with your personal information, including marking off the check boxes at the bottom of the screen. When you are done, click the green register button.



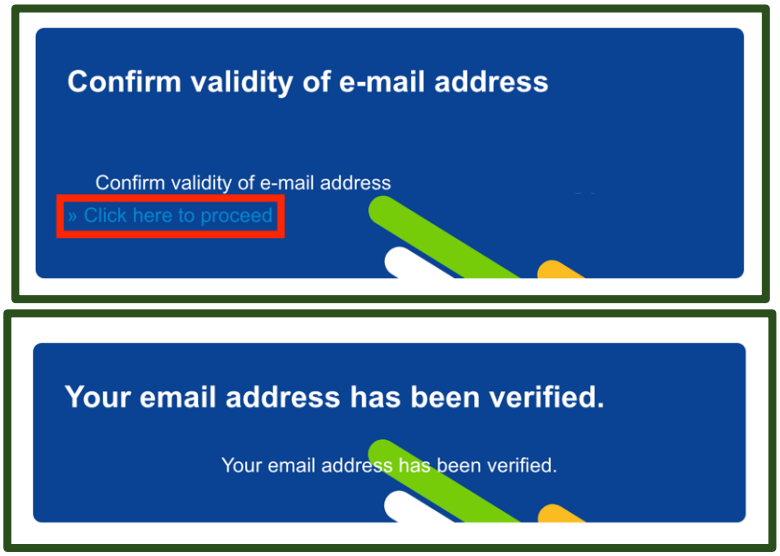
4. The window will now display **Thank you for signing up** and prompt you to verify your email. Open your email and look for the verification email.



5. Click the blue **Verify email address** button which will direct you back to the Copernicus Data Space website.



6. In the new window, click where it says **Click here to proceed**. It will then let you know that your email address has been verified.

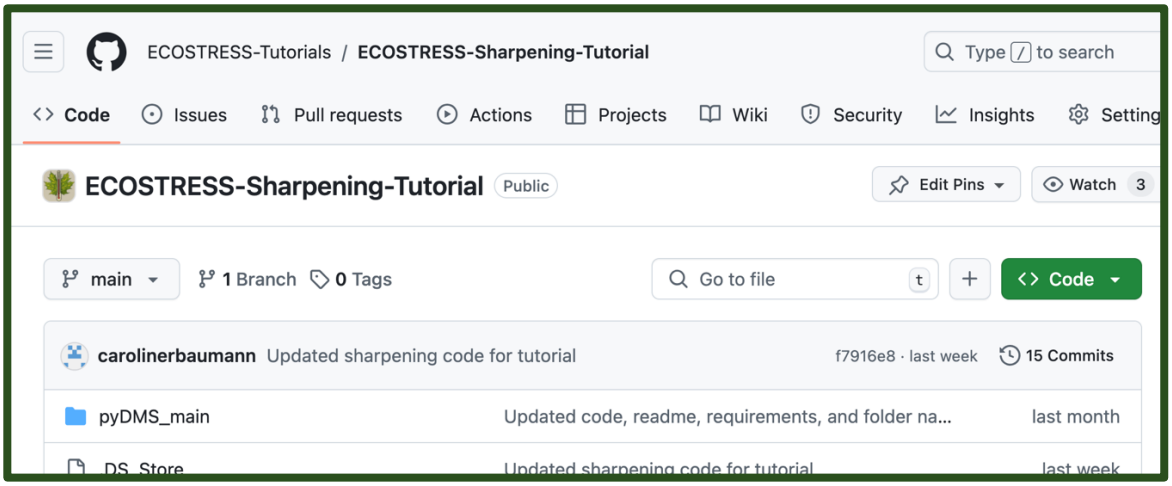


What is pyDMS?

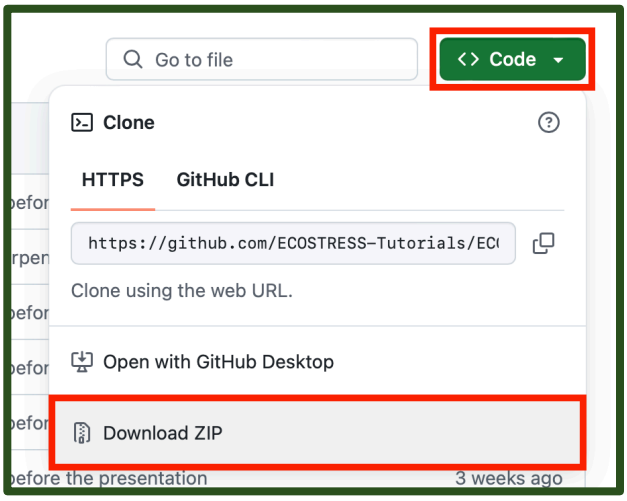
pyDMS is a Python library that implements the Data Mining Sharpened (DMS) algorithm, that is used to sharpen low resolution satellite imagery using high resolution data. We want to use this in our code, so we need to download and install it. We can download it from GitHub, which is an online platform used to store and share code.

DOWNLOADING CODE AND PYDMS FROM GITHUB

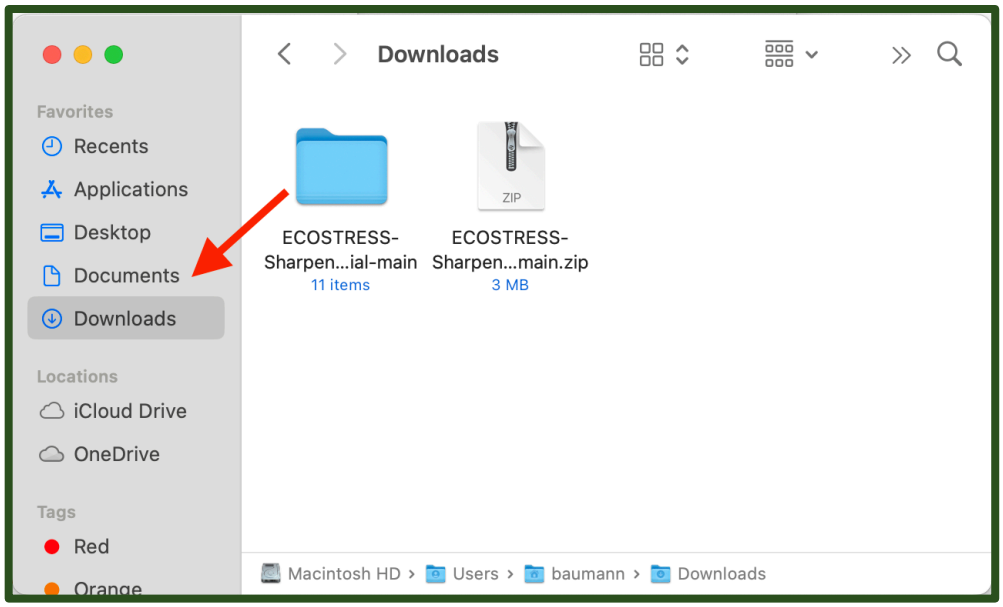
1. To access the pyDMS package and the code used in this tutorial, go to https://github.com/ECOSTRESS-Tutorials/ECOSTRESS-Sharpening-Tutorial.



2. At the top right, click the green button that says **<> Code**. In the dropdown select **Download ZIP**. A zip file containing everything in the GitHub repository will begin downloading.



3. Once the zip file has been downloaded, **double click** on it to un-zip it. This new folder will now function as your **project folder**. You can move it wherever you would like, but I am going to move mine to my documents.

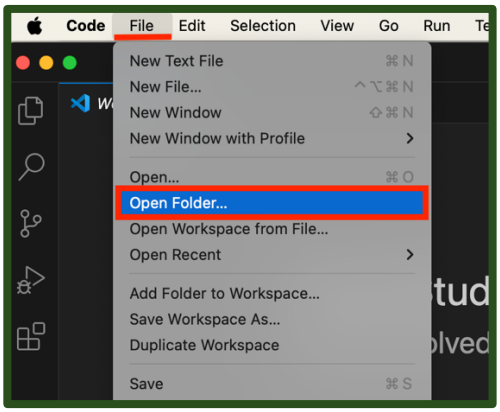


What is an OAuth client?

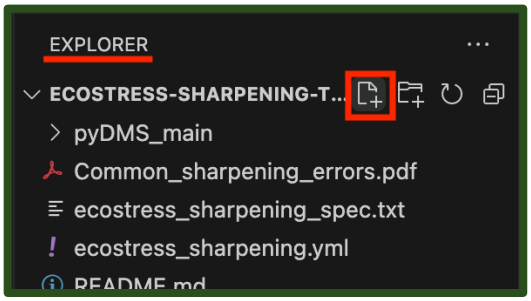
An OAuth client requests access to data on behalf of the user without needing their password. Instead, OAuth creates a secure token, or temporary key, that can be used to access the data for as long as you allow it. This ensures that your account details stay safe when downloading data.

CREATING A NEW OAUTH CLIENT

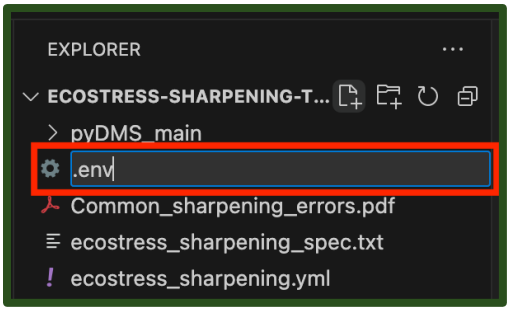
1. Open Visual Studio Code and get connected to your project folder by selecting **File > Open Folder.** In the pop-up finder window, select your project folder and click **Open**.



2. In the **EXPLORER** tab on the left, hover over the project folder and click the **new file icon**.

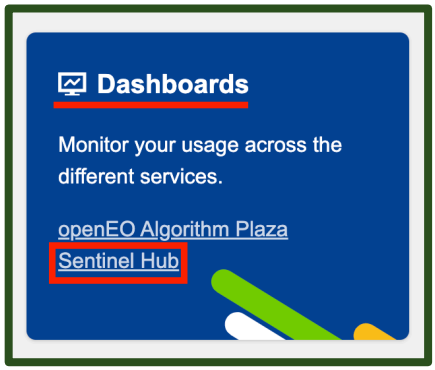


3. Name this new file **.env** and press enter.

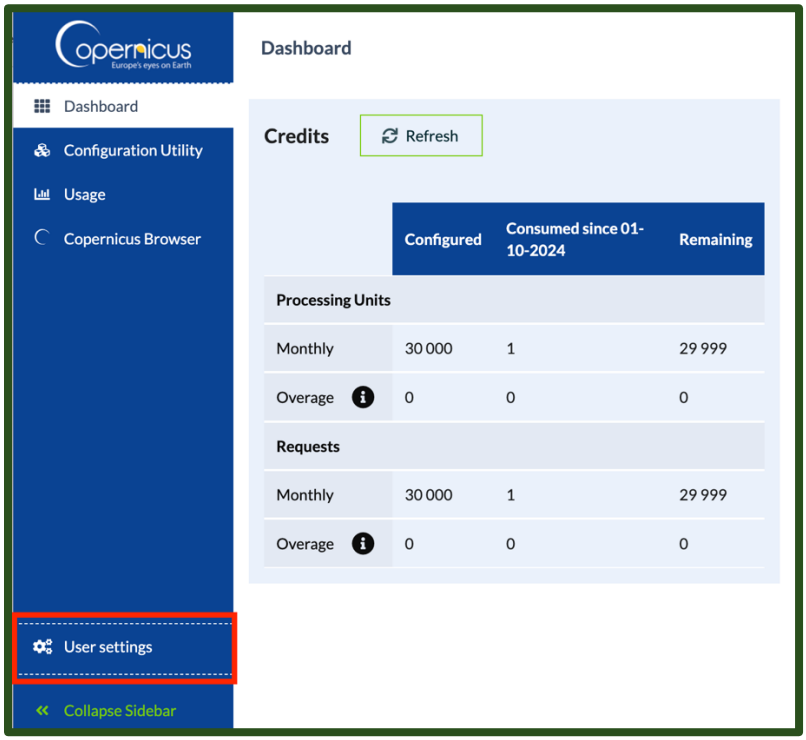


4. Next, in a browser window, go to https://dataspace.copernicus.eu/ or by search for **Copernicus Data Space** and log in. Then, click the **green profile icon**.

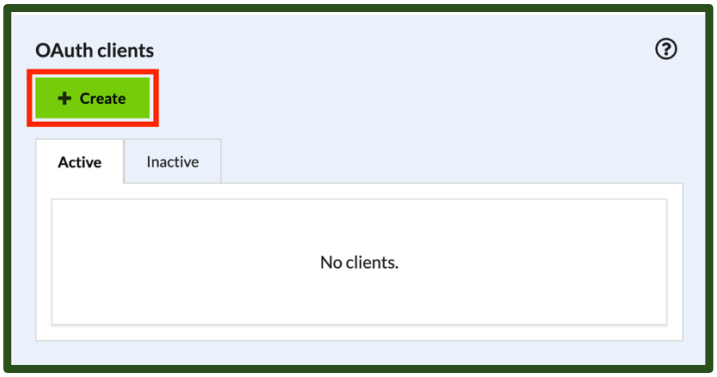
5. In the new window, find the box that says **Dashboards** and click the link that says **Sentinel Hub**.



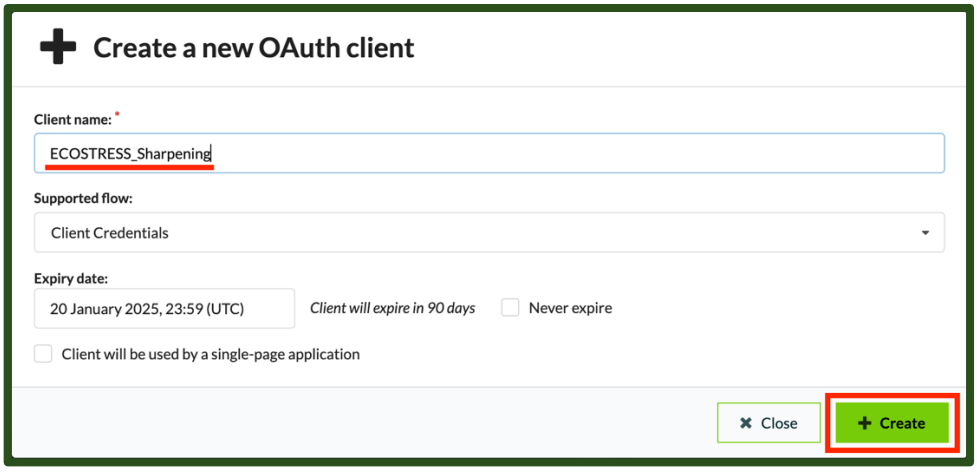
6. In the Dashboard window, click **User Settings** in the bottom right.



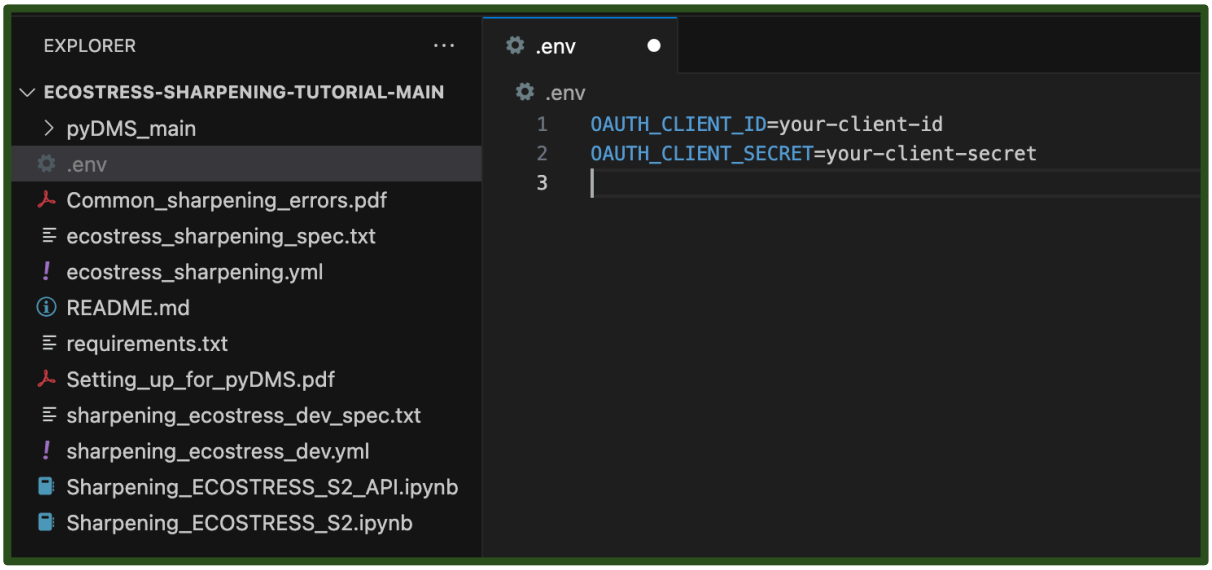
7. Look for the box titled **OAuth clients** and click the green **+Create** button.



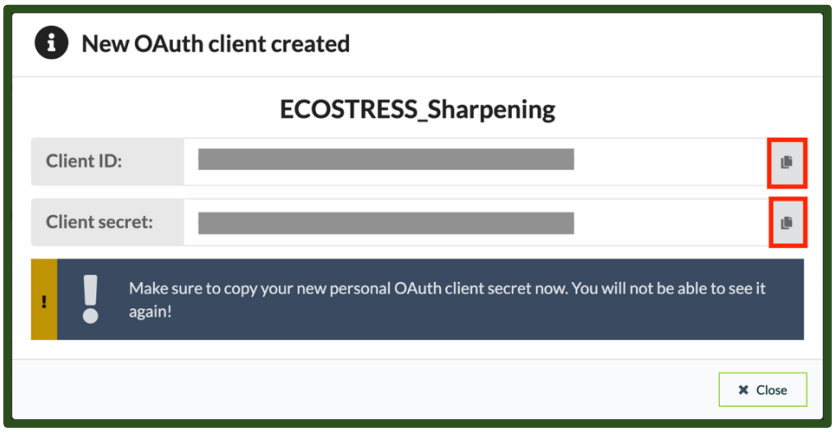
8. In the pop-up, type in a **Client name**. This name is just a way to identify the client for your organization and clarity. For example, I am going to name mine **ECOSTRESS_Sharpening**. Once you have entered a name, press the green **+Create** button.



9. A pop-up will appear with your Client ID and Secret. <u>Do not close this window because you will not be able to view it again once it is gone!</u> Navigate back to Visual Studio Code and open the **.env** file that you created. In the **.env** file, type:
    a. OAUTH_CLIENT_ID=your-client-id
    b. OAUTH_CLIENT_SECRET=your-client-secret

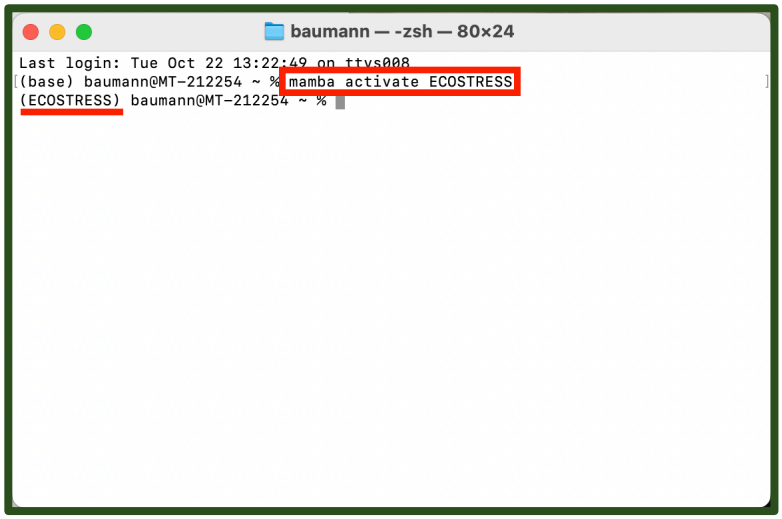


10. Replace **your-client-id** with the Client ID that was given in the Copernicus Data Space OAuth creation by copying and pasting. Do the same with **your-client-secret**. Save your .env file. You can now close the OAuth pop-up window.

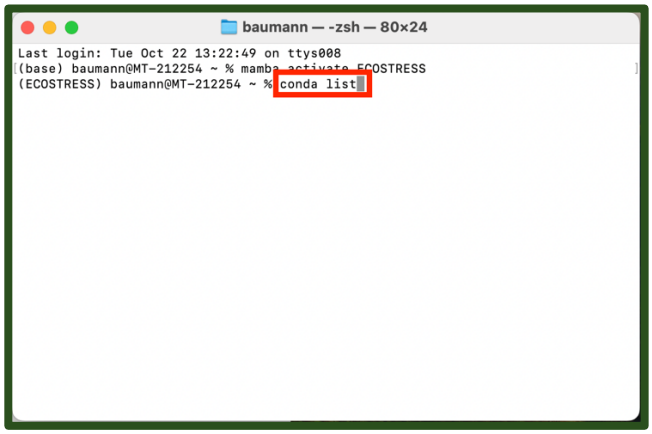


HOW TO INSTALL THE REQUIRED PACKAGES FOR YOUR ENVIRONMENT

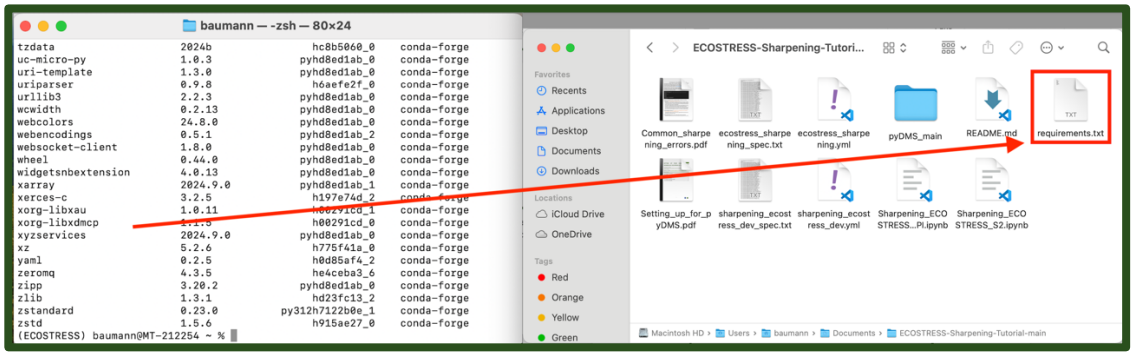
1. If you followed the creating an environment tutorial, you will need to install a few more packages to the ECOSTRESS environment you created. If you are working with a different environment, or using the ECOSTRESS environment from a previous tutorial, you can look at the different packages installed within your environment to see what you have and what you need.
    a. To do this, open the **terminal** and type **mamba activate** followed by the name of your environment. Press enter to run. You will know your environment has been activated when its name shows up in parentheses.



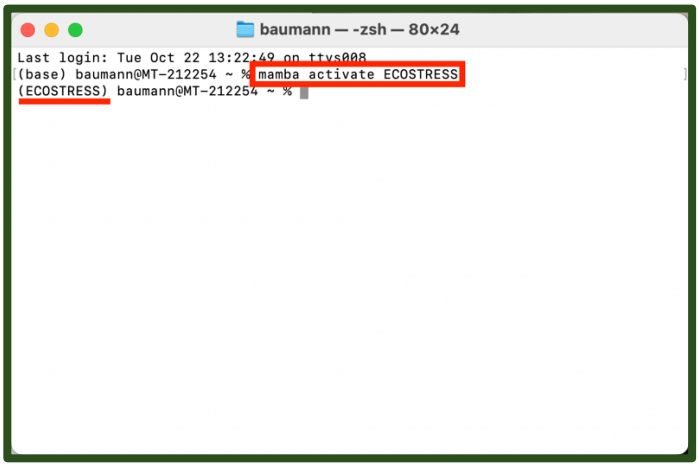
b. Then type **conda list** and press **enter** to run. This will list all the packages in your environment.



c. Compare this to the list of packages on the **requirements.txt** document that you downloaded from the GitHub as part of the main project folder. Take note of which ones you still need to install.

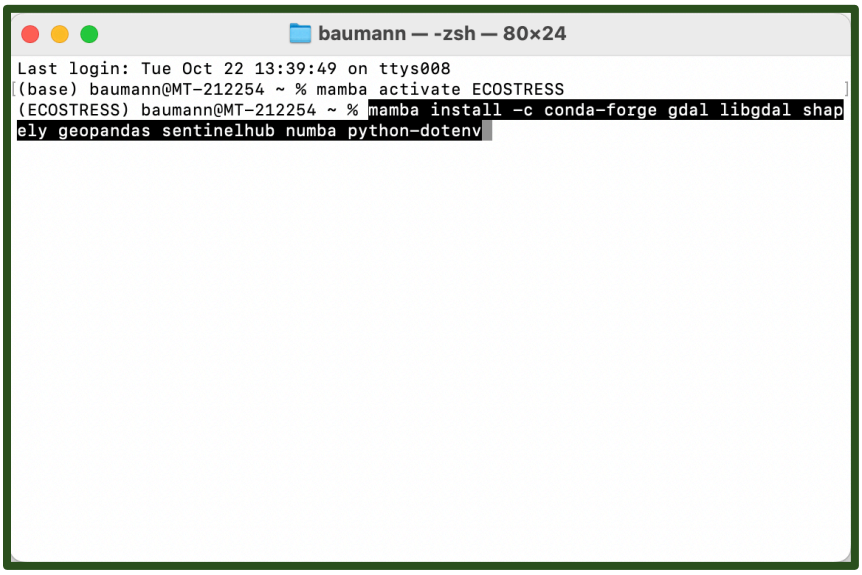


2. To install the remaining packages, first make sure that your environment is activated (its name should be listed at the start of the terminal command line in parentheses). If it is not activated, type **mamba activate** followed by the name of your environment and run it.



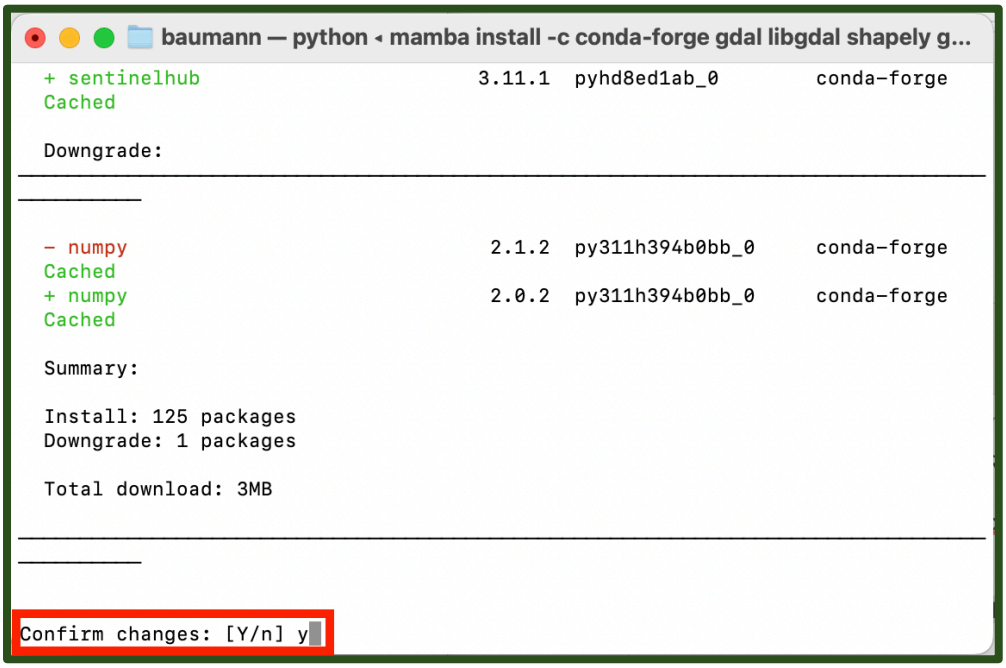
3. Then, type **mamba install -c conda-forge** followed by the name of all the packages you need to install. If you used the Creating an Environment ECOSTRESS tutorial, you can copy and paste this into the terminal and run it to get all the remaining packages installed:

mamba install -c conda-forge gdal libgdal shapely geopandas sentinelhub numba python-dotenv

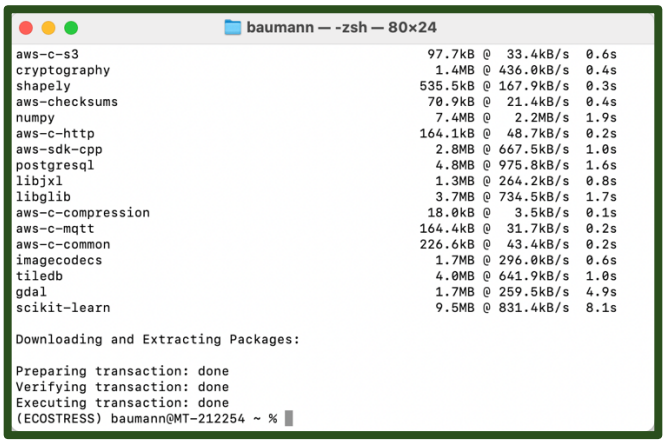


   a. However, it is best to list the packages in your own environment and make sure you are missing the same ones as me. If you are missing different ones, you can modify the command accordingly.

4. It may ask you to **Confirm changes y/n** for which you can type **y** and press enter.

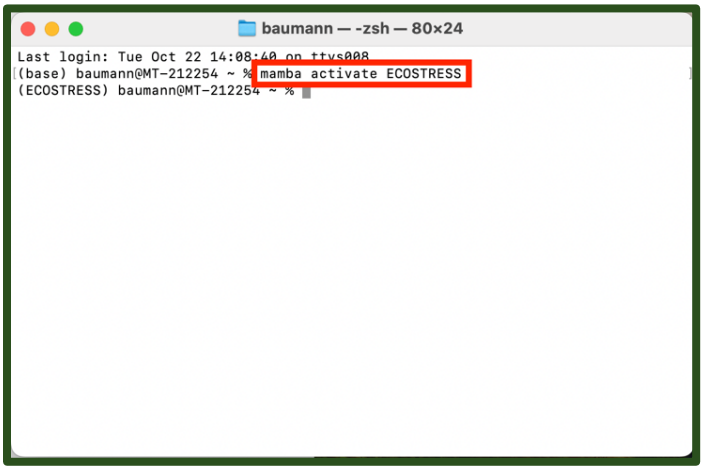


5. It should look something like this when it is done:



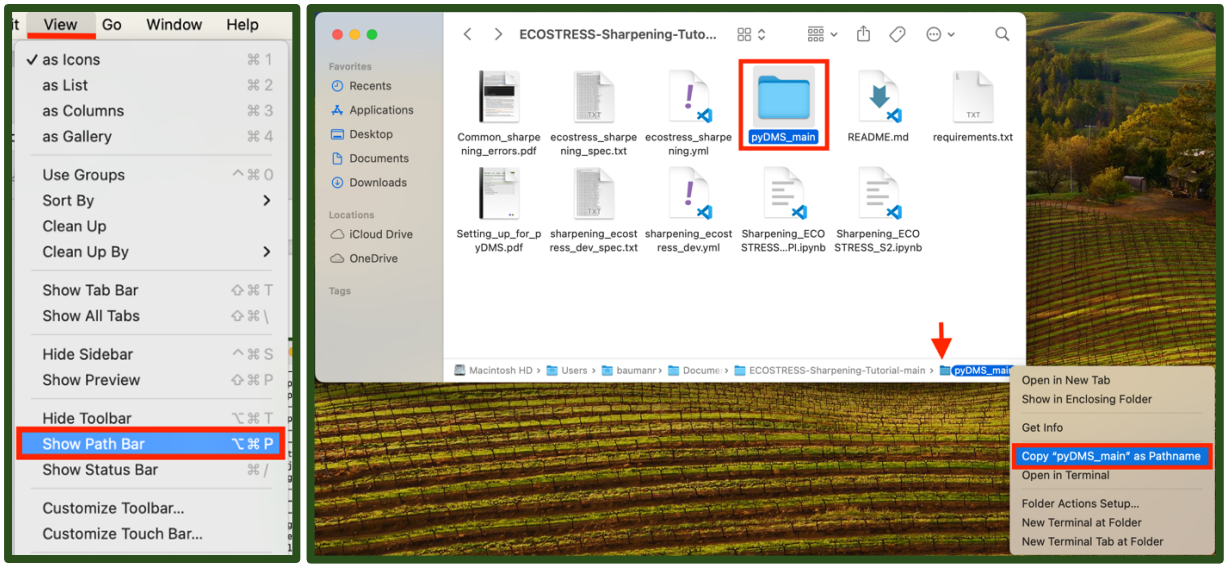
HOW TO INSTALL PYDMS

1. Open the terminal and activate your environment by typing **mamba activate** followed by the name of your environment.

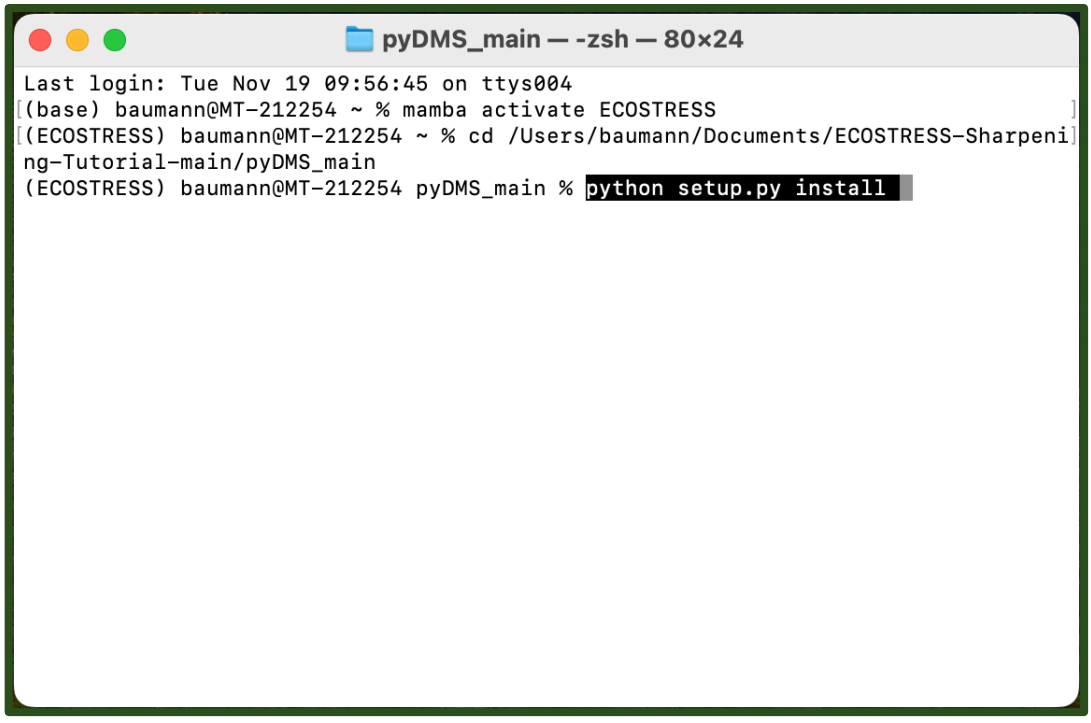


2. Then, change the directory to the pyDMS_main folder by typing the command **cd** followed by a space and the path to the folder.

a. To copy the path to the folder, go to **View > Show Path Bar**. Then in your finder, navigate to the folder. Find where the folder is listed in the path bar on the bottom of the window. Right click it and select **Copy "pyDMS_main" as Pathname**.

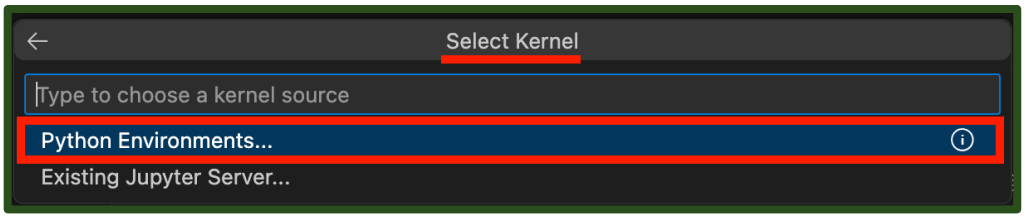


3. Then, in the terminal type **python setup.py install** and run it. Now you have an environment set up to run the downscaling code with.

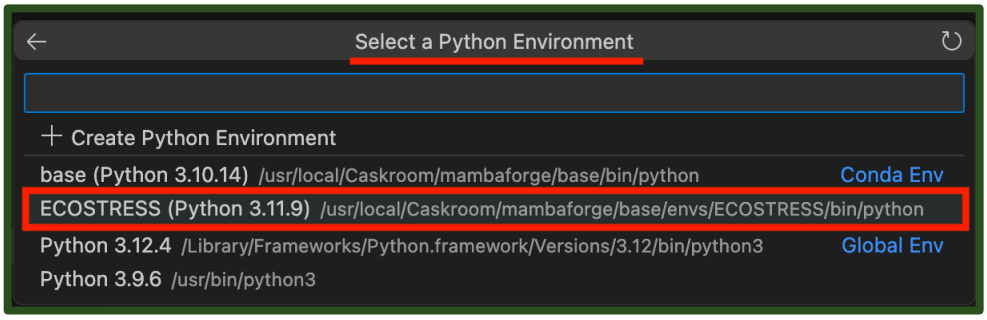


SETTING UP AND RUNNING THE CODE

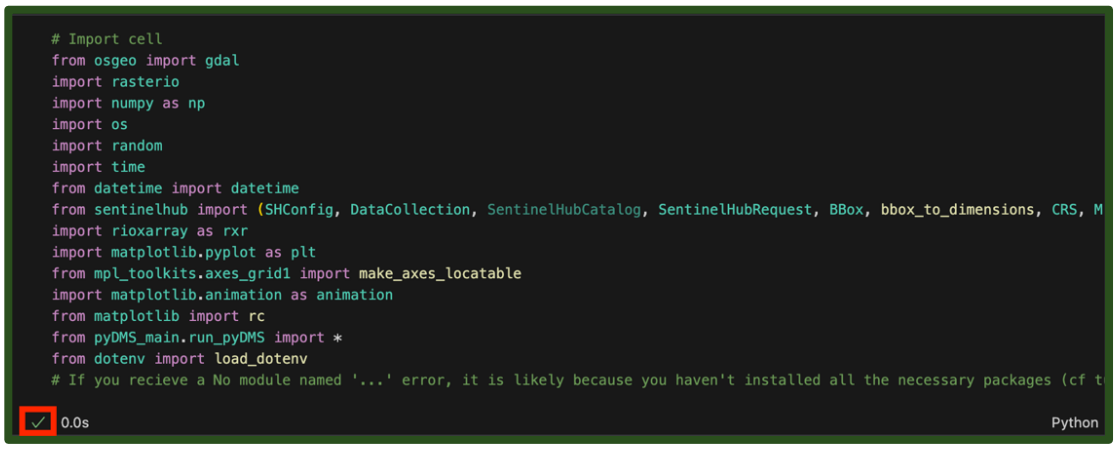
1. In **Visual Studio Code**, open the **Sharpening_ECOSTRESS_S2.ipynb** Jupyter Notebook. At the top of the file there is a lot of information about how the code works that you can read if you are interested. Scroll down to the block of code that is used to **import libraries**. Click into the code and press **Shift + Enter** to run it.
    a. At the top of the window, a pop up will appear prompting you to **select a kernel** to run your code with. Click on **Python Environments ...**



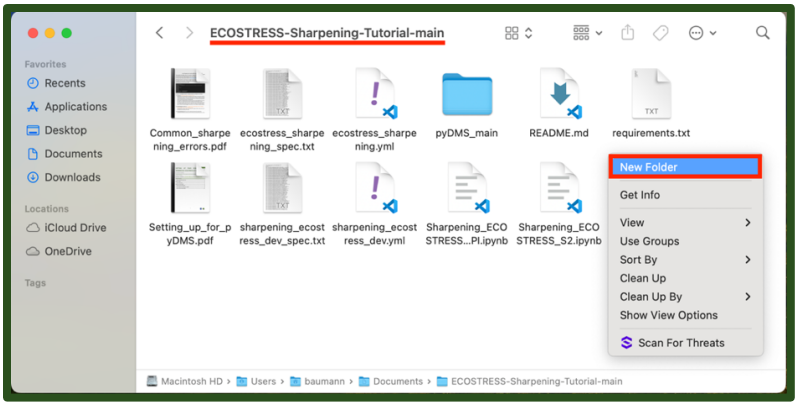
    b. Select the **ECOSTRESS** environment that you created, or another one if you have a different one you want to use.



    c. You will know it is done running when a green check mark appears on the bottom left of the cell.



2. The next section of code sets up all the **directories** (dir) to input and output locations. In the **ECOSTRESS-Sharpening-Tutorial-main** folder, I suggest making separate folders for each of the required directories. You can do this by **right clicking** in the **Finder** window, selecting **New Folder**, and naming the folder.



3. The folders that you need include:

   a. **S2_output_folder** – a folder to store the Sentinel-2 images that the API will download. Leave this empty for now.
   b. **QC_dir** – a folder with all of the QC files that you downloaded. If you do not have QC files, you do not need this folder.
   c. **cloud_dir** – a folder with all of the cloud mask files that you downloaded.
   d. **lst_dir** – a folder with all of the raw LST files that you downloaded
      i. Alternatively, if you already scaled your LST files, you can put them in a folder called **lst_dir_sc**
   e. **dst_dir** – a destination folder where your sharpened images will be saved.
4. Next, **copy** the **path** to these folders and **paste** them into the code next to their associated variable.

Example:

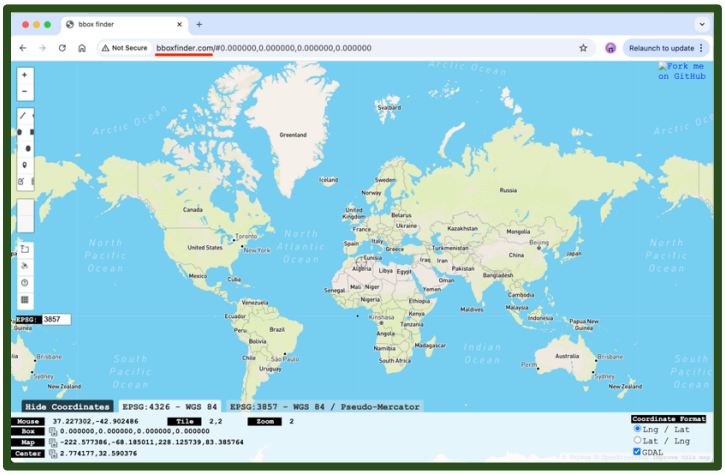
```
# Choose your output folder for the downloaded Sentinel-2 products
s2_output_folder = r'/Users/baumann/Documents/ECOSTRESS-Sharpening-Tutorial-main/s2_output_folder'
# Folder where all the ECOSTRESS Quality Control files are located. This can remain empty if you don't have any QC file.
QC_dir = r'/Users/baumann/Documents/ECOSTRESS-Sharpening-Tutorial-main/QC_dir'
# Folder where all the ECOSTRESS Cloud mask files are located. It is required to download and the the cloud mask if you are
cloud_dir = r'/Users/baumann/Documents/ECOSTRESS-Sharpening-Tutorial-main/cloud_dir'
# Folder where all the ECOSTRESS LST files are located for the scene of interest.
lst_dir = r'/Users/baumann/Documents/ECOSTRESS-Sharpening-Tutorial-main/lst_dir'
# If your ECOSTRESS LST files are already scaled, then you can type here the directory in which they are located. If they a
lst_dir_sc = r''
# Folder where all the sharpened ECOSTRESS LST files will be written for the scene of interest
dst_dir = r'/Users/baumann/Documents/ECOSTRESS-Sharpening-Tutorial-main/dst_dir'
```

5. Next, scroll down to the section of the code under **OAuth Copernicus Data Space.** If you followed this tutorial and set up your OAuth in a **.env** file, you should be able to press **Shift + Enter** to run this block of code. If you set up your OAuth in another way, you may need to adjust the code accordingly.

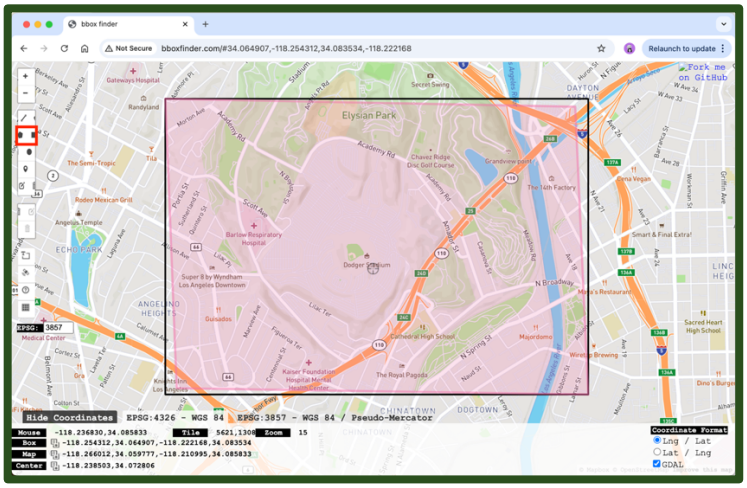
```
config = SHConfig()
config.sh_base_url = 'https://sh.dataspace.copernicus.eu'
config.sh_token_url = 'https://identity.dataspace.copernicus.eu/auth/realms/CDSE/protocol/openid-connect/token'
load_dotenv()
config.sh_client_id = os.getenv('OAUTH_CLIENT_ID')
config.sh_client_secret = os.getenv('OAUTH_CLIENT_SECRET')
```

✓ 0.0s Python

6. Scroll down to the next block of code under **Set the parameters for Sentinel-2 data to be downloaded** and find the variable titled **aoi_coords_wgs_84**. We need to set this variable equal to the coordinates of a bounding box for our study area. To get these coordinates, click on the link in the code or search the web for **bboxfinder.com**.

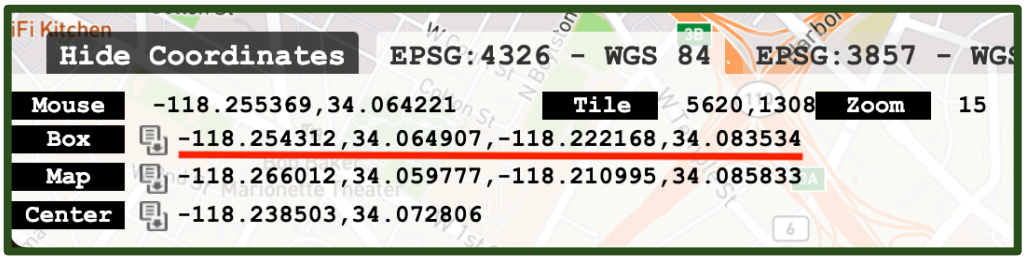


7. On the website, **zoom into** your area of interest on the map. Then click the **draw a shape icon**. Click on the map to create a box around your area of interest and click on the first point you made to close the box.



8. Then, at the bottom of the screen, **copy** the coordinates listed after **Box**. Go back to Visual Studio Code and **Paste** these coordinates into the variable.

Example:




```
# The coordinates of the bounding box of your choosing, in lat lon : (xmin,ymin,xmax,ymax)
# Use the http://bboxfinder.com to find your box easily (already in the right order)
aoi_coords_wgs_84 = (-118.254312,34.064907,-118.222168,34.083534) # example
```

9. Next, you can set the resolution of the Sentinel 2 data by adjusting the variable called **s2_res**. For now, I will leave it at 20.

```
# Choose the
s2_res = 20
```

10. Finally, we need to change the **interval** variable to represent the start and end date for which we are interested in getting data. Make sure to enter these dates in the **"YYYY-MM-DD"** format.

Example:

```
# Choose your time interval (beginning, end) in the
interval = ("2024-06-01", "2024-08-31") # example
```

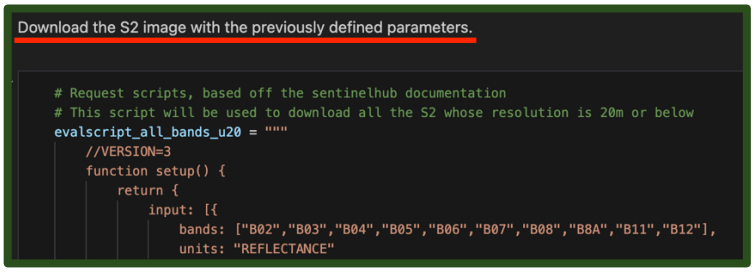
11. Once all of the variables are set, run this block of code by pressing **Shift + Enter.**

```
# The coordinates of the bounding box of your choosing, in lat lon : (xmin,ymin,xmax,ymax)
# Use the http://bboxfinder.com to find your box easily (already in the right order)
aoi_coords_wgs_84 = (-118.254312,34.064907,-118.222168, 34.083534) # example

# Choose the resolution of the Sentinel data in meters 10 or 20, this will be the final resolution of the downscaled image.
s2_res = 20

# Render the coordinates usable by the Copernicus API
aoi_bbox = BBox(bbox = aoi_coords_wgs_84,crs=CRS.WGS84)
aoi_size = bbox_to_dimensions(aoi_bbox,resolution = s2_res)

print(f"Image shape at {s2_res} m resolution: {aoi_size} pixels") # The size of the box is limited to 2500 pixels in each d
if (aoi_size[0]>2499 or aoi_size[1]>2499) :
    raise(ValueError("The box is limited to 2500 pixels in each direction, try again with a smaller bounding box."))


# Choose your time interval (beginning, end) in the format (YYYY-MM-DD). You'll receive a S2 image from the tile with the l
interval = ("2024-06-01", "2024-08-31") # example
```

0.0s Python

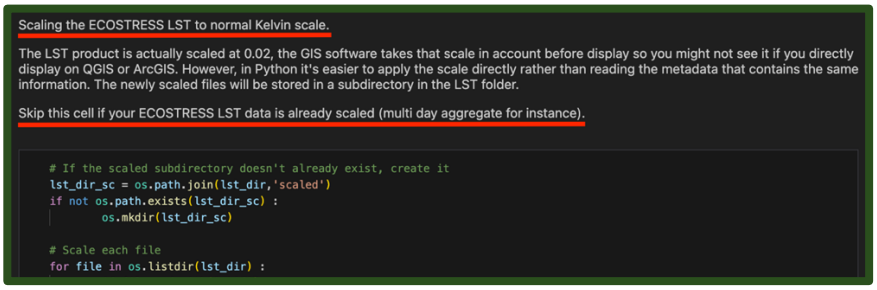
```
Image shape at 20 m resolution: (150, 101) pixels
```

12. Run the next **three** blocks of code including:
    a. **Download the S2 image with the previously defined parameters.**

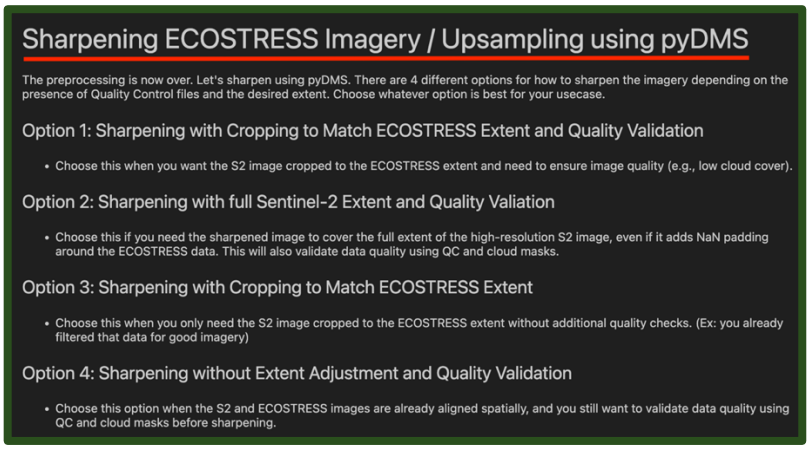


    b. **Preprocessing the QC files**

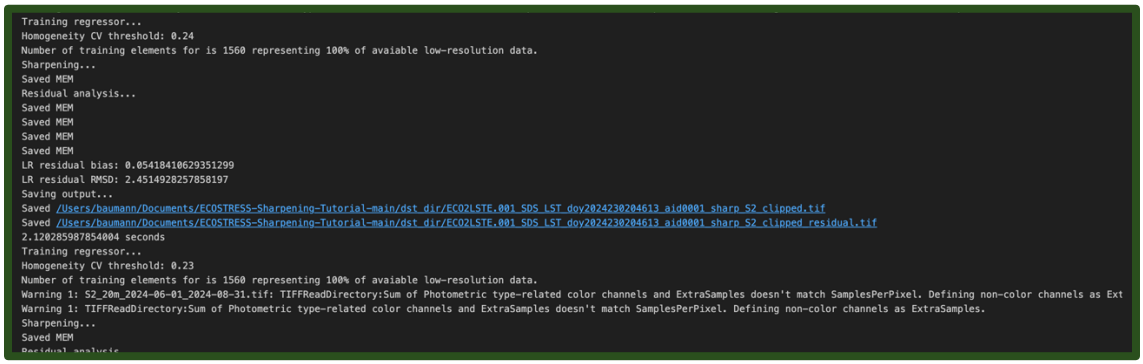
    c. **Scaling the ECOSTRESS LST to normal Kelvin scale.** However, if you used pre-scaled images, you can skip running this block of code.



13. You are now ready to sharpen your images! There are **four different options** of code blocks that you can run depending on your desired **extent** and if you have **QC files** or not. Read this section of the code and **decide which option is best** for your data:



14. Once you have decided which block of code is best for your data, scroll down to that section, **click** into the code block, and press **Shift+Enter** to run it. You should get a list of outputs that looks something like this:



15. You now have sharpened ECOSTRESS imagery! In order to see an **image plotted** of the sharpened scenes, you can run the block of code under the **Display** section.

a. Example of **Plot one random sharpened image:**