

Using the GEE (Google Earth Engine) API to access dynamic world data (or other types of data)

Visualize DW data here:

<https://www.dynamicworld.app/explore/>

Here is the paper describing the methods:

<https://www.nature.com/articles/s41597-022-01307-4>

you'll need to register to be able to access the data and the API (the tool used for accessing the data)

To explore the data, first go to:

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

Go to "Datasets" (top row tab, you'll need to be registered probably)

In the Search tab, type in "Dynamic World data"

Select "Dynamic World V1 | Earth Engine Data Catalog | Google for ..."

- you can see the date range for the data in the "Dataset Availability" section. It should start in June of 2015

- you'll see an explanation of the different bands/land type classifications here too. This will be important later.

You should see some example JavaScript and Python script here too. I'm using JavaScript for now (seems easier)

***to access the API and pull some data

Go to: code.earthengine.google.com

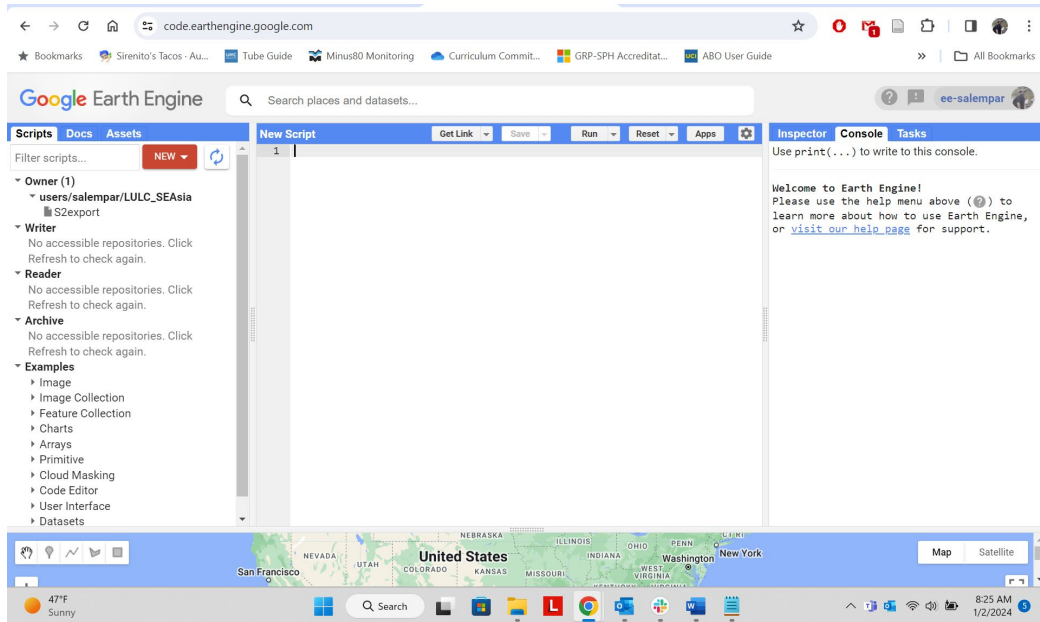
Here you can begin working with JavaScript to work with the relevant data.

Parker Group Tutorial

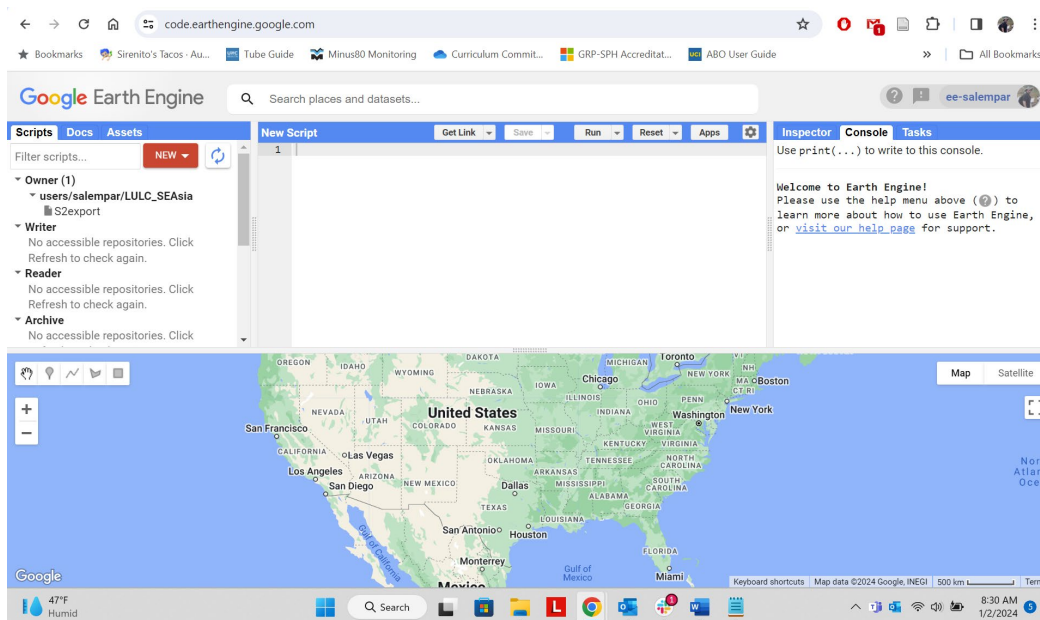
January 1, 2024

Go to: code.earthengine.google.com

Your screen should look similar to this:



//Notice that there is a map at the bottom of the API. You can extend that map so that you can zoom in on different areas

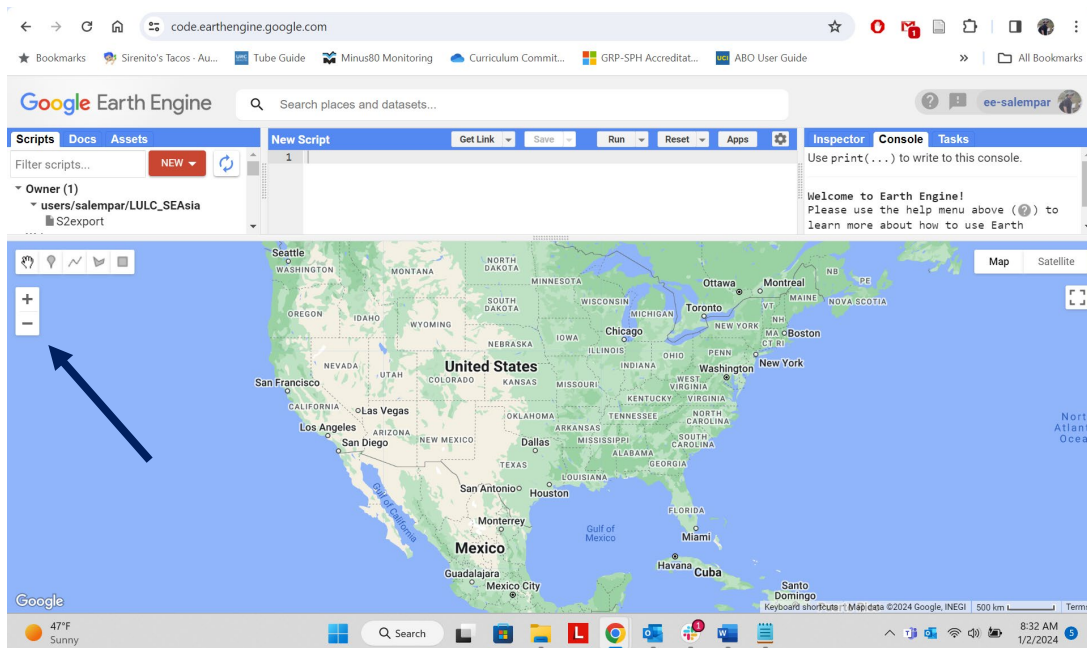


//We need to define an area for which we will download the LULC data. I will name mine "MaeTan2" and it will be defined by a drawn polygon

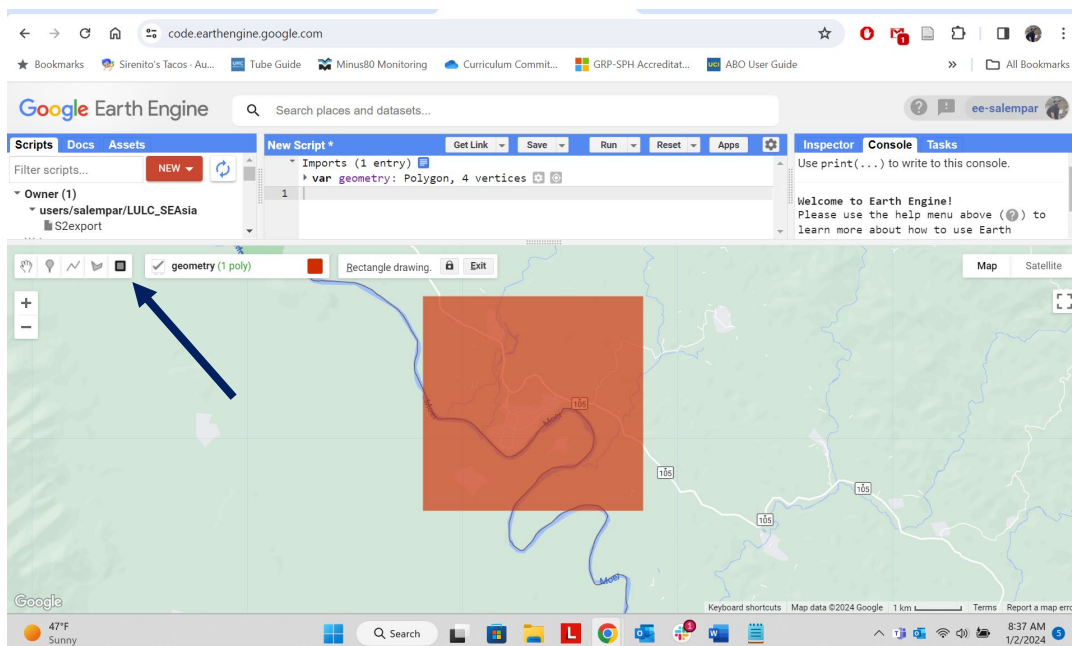
Parker Group Tutorial

January 1, 2024

//Scroll to the area you want to use for downloading data and zoom in using the tools in the map. It will be simpler to start with a small area (downloaded data for large areas will be very large!!)



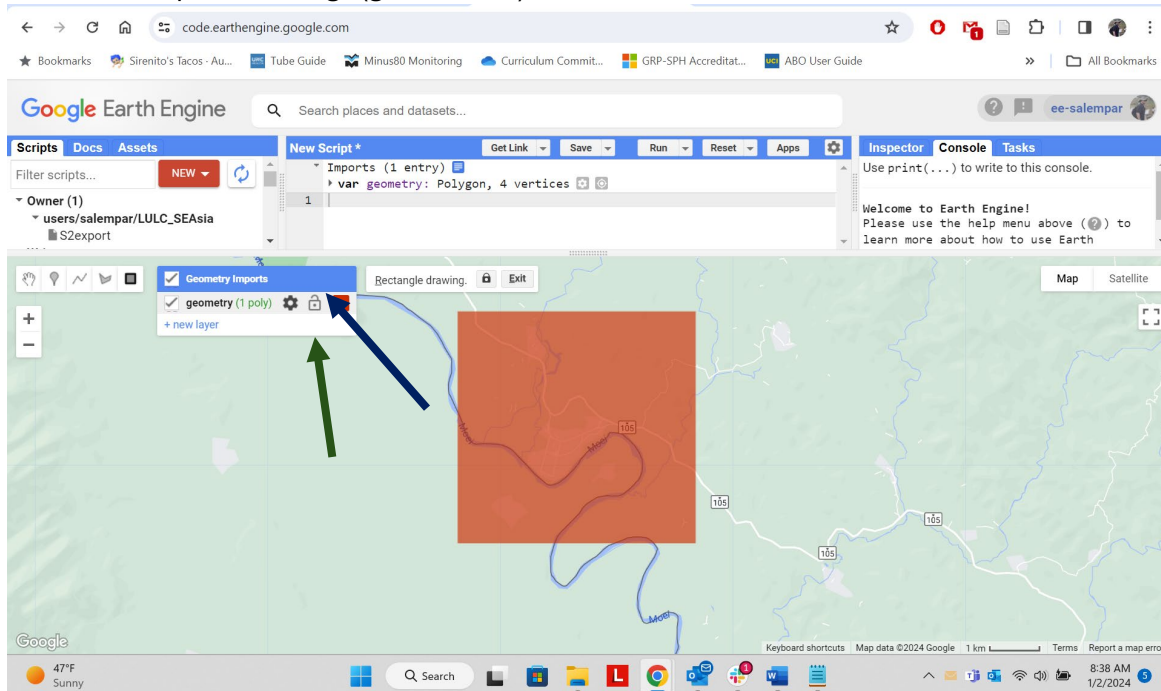
//Once you've zoomed in on your chosen area, you can select it using one of the drawing tools in the map



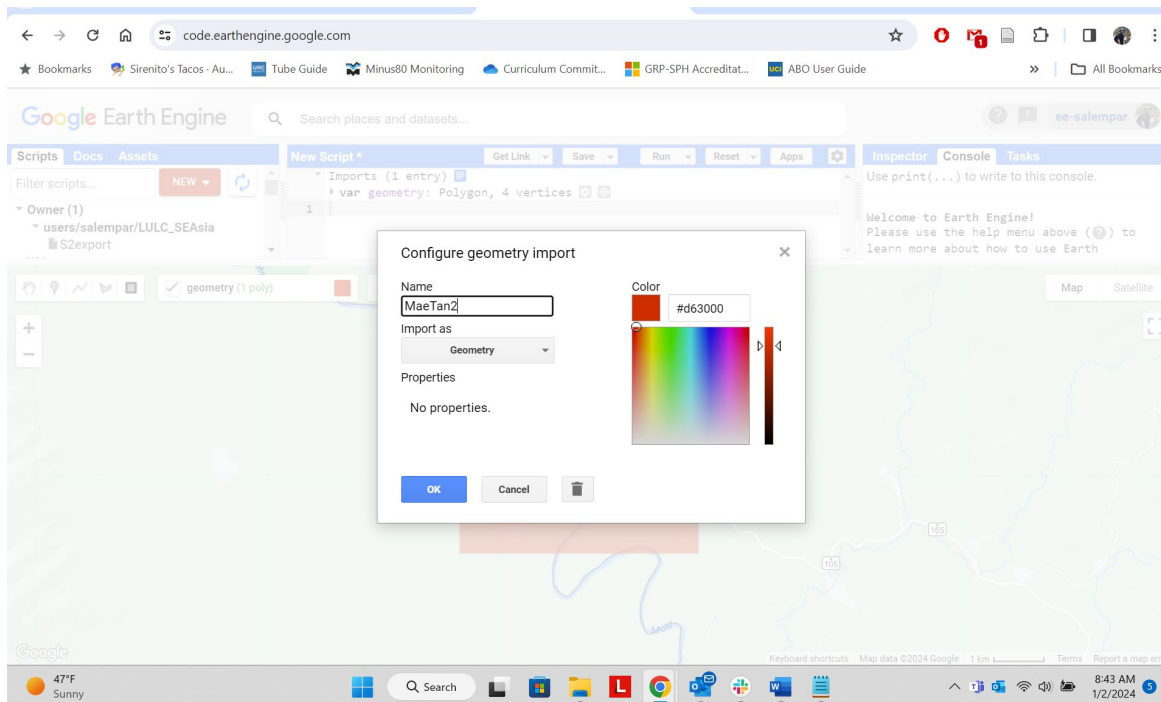
Parker Group Tutorial

January 1, 2024

//place your cursor over the “Geometry Imports” tab on the map (blue arrow below) and you should
// see a new level emerge under it (labeled “geometry(1poly)” here. Place your cursor over the star-
// like shape for settings (green arrow)



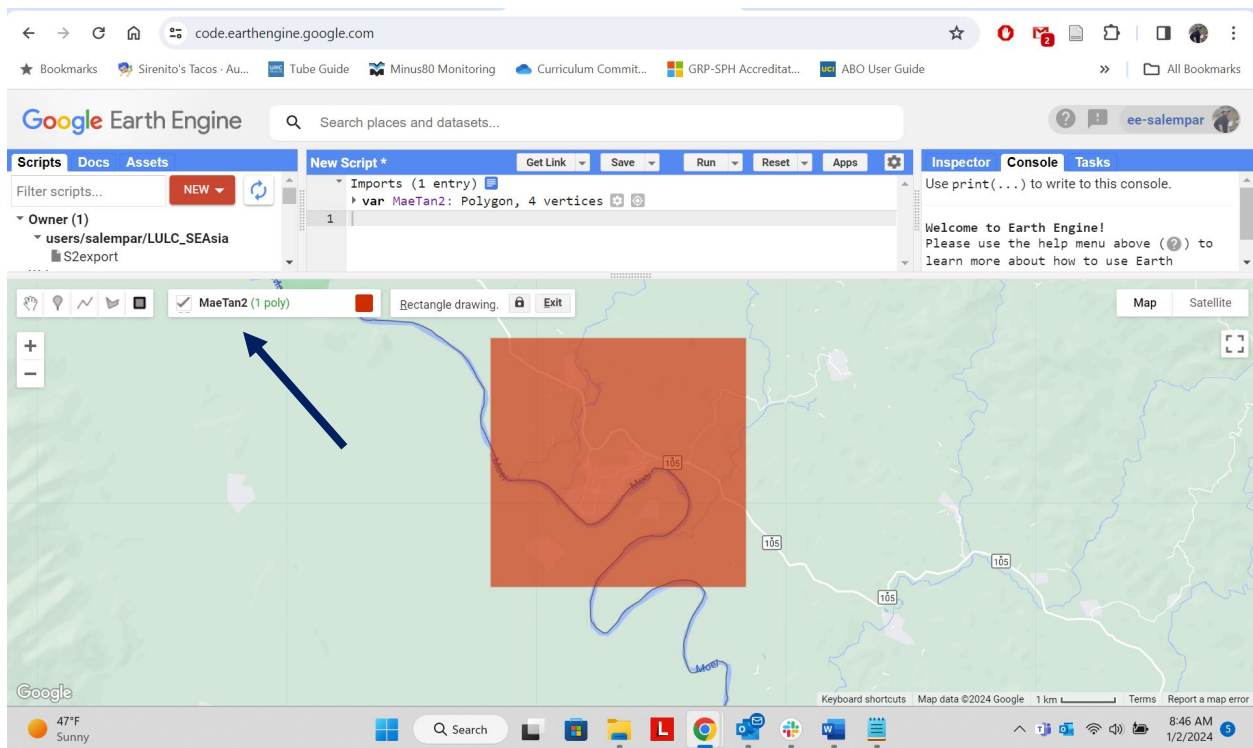
//You should now see a new box emerge on your screen, and in that box you can provide a custom
// label for your area. I’m using “MaeTan2”. Note this name in the following JavaScript text.
// Click on the blue “OK” button.



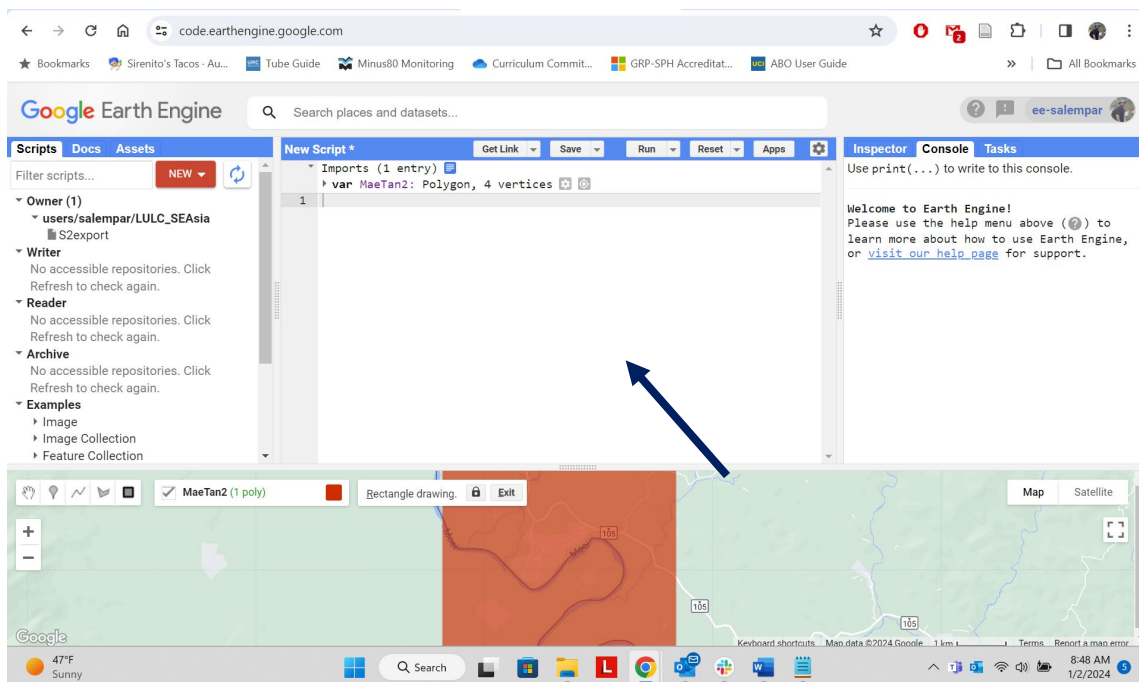
Parker Group Tutorial

January 1, 2024

// you should now see that the shape has a name in the map



// Now we are going to move to the script part of this process. It will help if you minimize the map
// we will be working in the part of the API with the blue arrow below

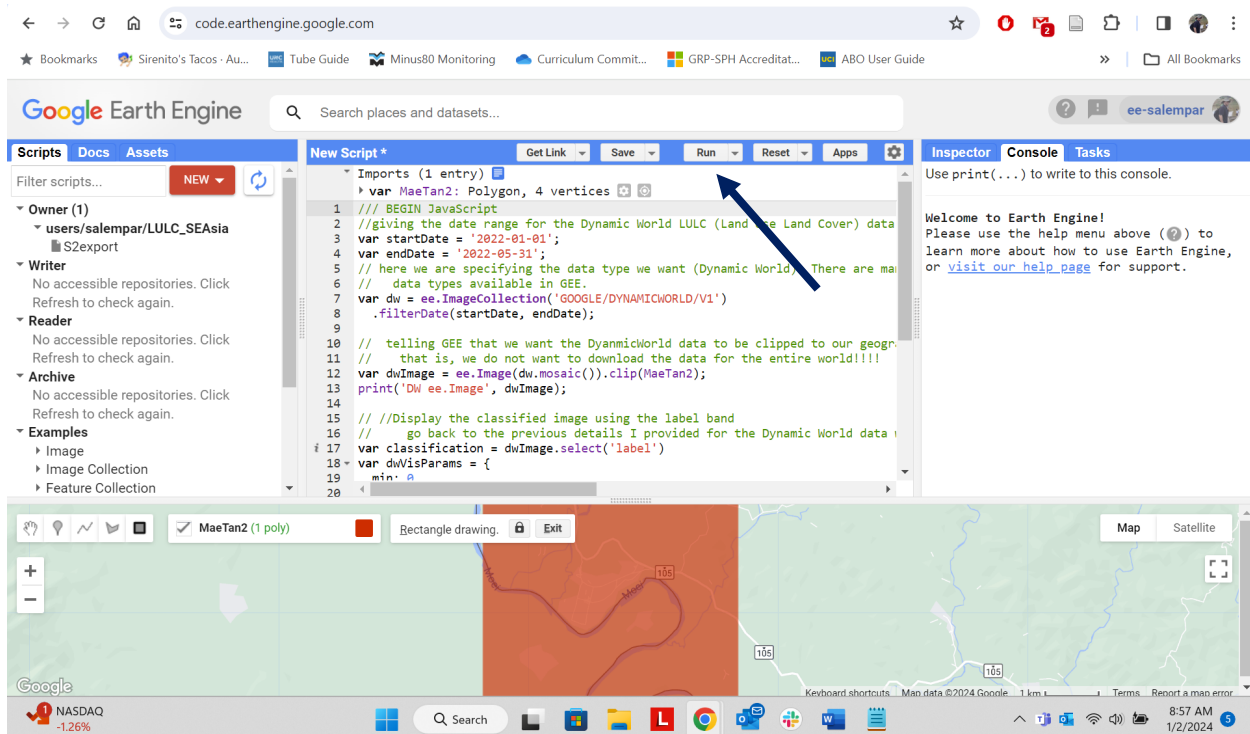


Parker Group Tutorial

January 1, 2024

//You will need to update the parts of the following script that say “MaeTan2” with whatever you’ve
// named your geographic area of interest. Then you should be able to just copy and paste the
// JavaScript into the API. It will of course be useful for you to go through the script to make sense
// of what it is doing.

//After you’ve pasted the JavaScript into the API window, you’ll need to click on “Run” (blue arrow
// below). If there are errors in your code, it will show up in the window to the right, including the
// line that has an error. This way you can debug line-by-line if necessary.



```
/// BEGIN JavaScript
//giving the date range for the Dynamic World LULC (Land Use Land Cover) data that we will pull
var startDate = '2022-01-01';
var endDate = '2022-05-31';
// here we are specifying the data type we want (Dynamic World). There are many, many different
// data types available in GEE.
var dw = ee.ImageCollection('GOOGLE/DYNAMICWORLD/V1')
    .filterDate(startDate, endDate);

// telling GEE that we want the DyanmicWorld data to be clipped to our geographic area of interest
// that is, we do not want to download the data for the entire world!!!!
var dwImage = ee.Image(dw.mosaic()).clip(MaeTan2);
print('DW ee.Image', dwImage);

///Display the classified image using the label band
```

Parker Group Tutorial

January 1, 2024

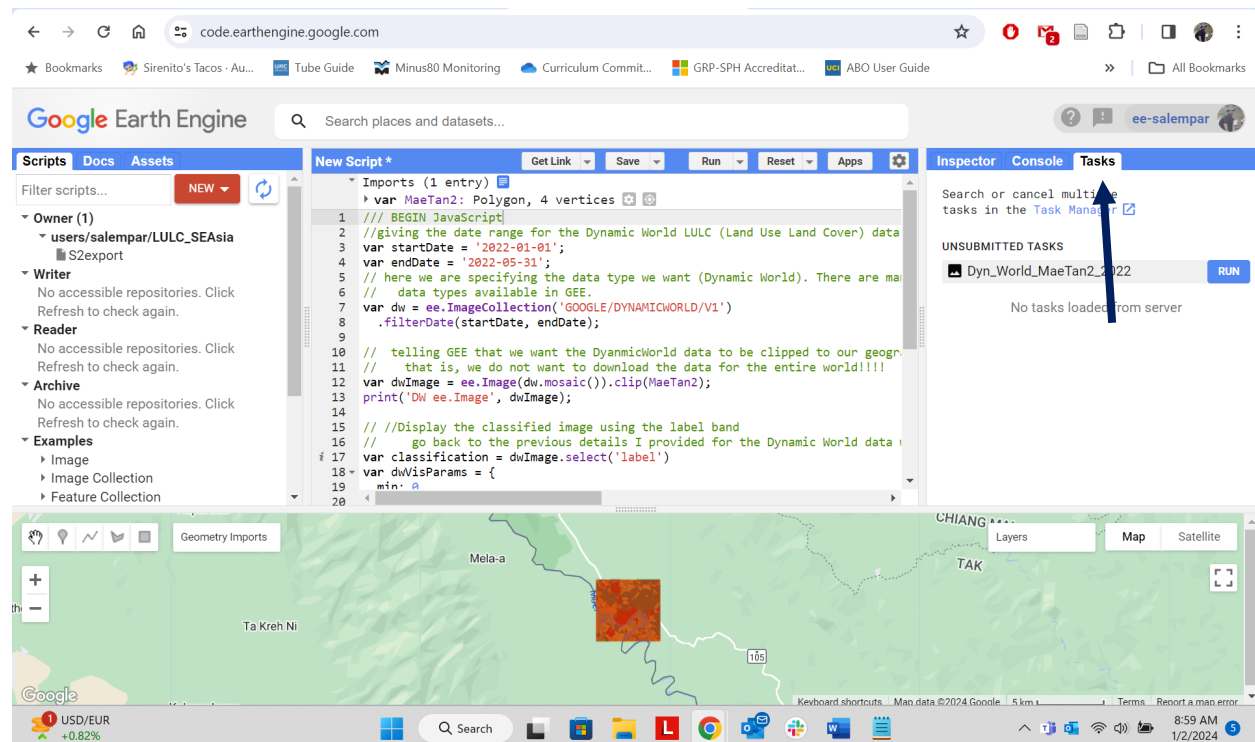
// go back to the previous details I provided for the Dynamic World data webpage. That page
// provides the different data bands and what they represent.

```
var classification = dwImage.select('label')
var dwVisParams = {
  min: 0,
  max: 8,
  palette: ['419bdf', '397d49', '88b053', '7a87c6', 'e49635', 'dfc35a', 'c4281b',
    'a59b8f', 'b39fe1']
};
Map.addLayer(classification, dwVisParams, 'Classified Image');
Map.centerObject(MaeTan2);
```

//We must first export these data to Google Drive

```
Export.image.toDrive({
  image: classification,
  description: "Dyn_World_MaeTan2_2022",
  scale: 10,
  region: MaeTan2,
  maxPixels: 1e13
});
```

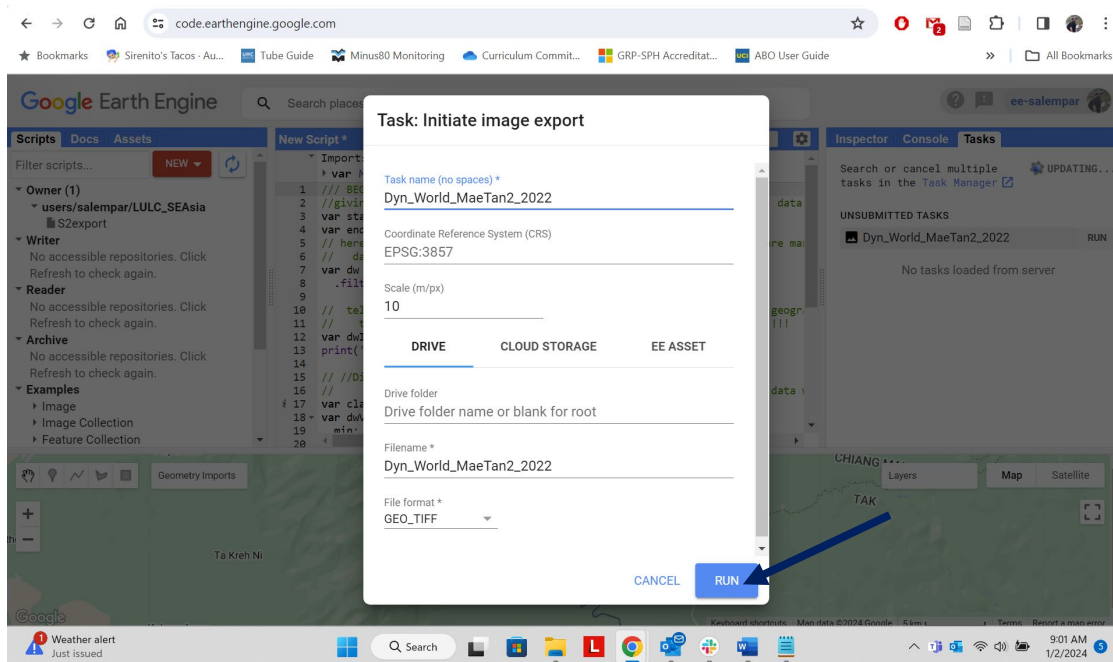
//NOW, on the top right hand side of the API, click on the tab labeled “Tasks” (blue arrow)
// you should see any unsubmitted tasks here, with the name you’ve provided for those tasks
// I suggest a name that will help future you (I’m using the data type, geog location, and year)
// next, click on “RUN” (green arrow)



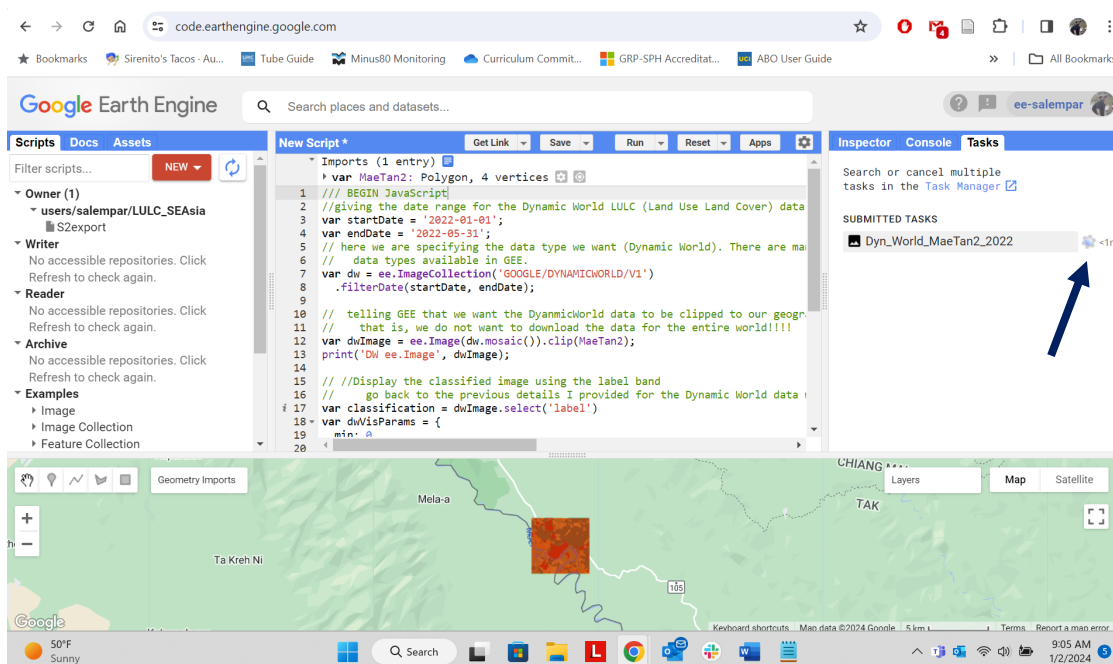
Parker Group Tutorial

January 1, 2024

// a new box should emerge. You shouldn't need to change anything here, unless you want to
// change the name for the file you'll be downloading.
// GEO_TIFF is a good file format, as it is a georeferenced raster that can be viewed in QGIS,
// ArcGIS, etc. Click the "RUN" button on the bottom right.



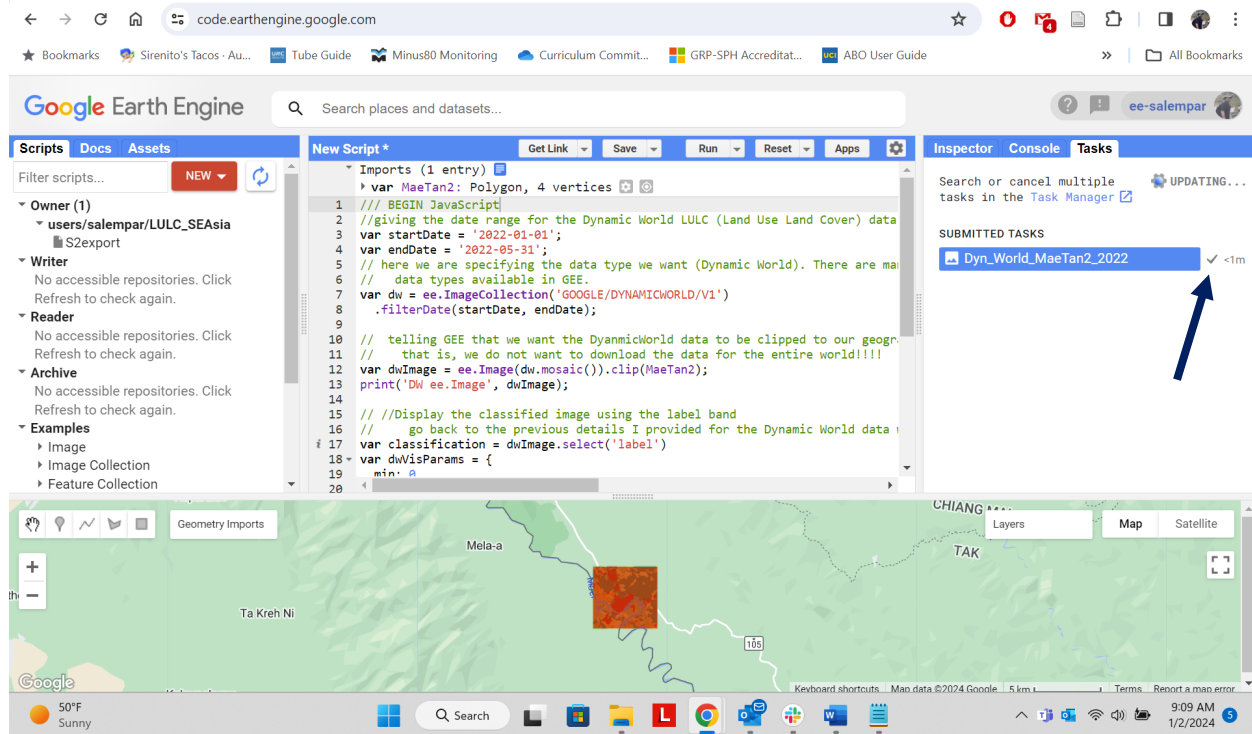
//Now you'll see a spinning wheel next to the submitted task (blue arrow), this means the process is
// working.



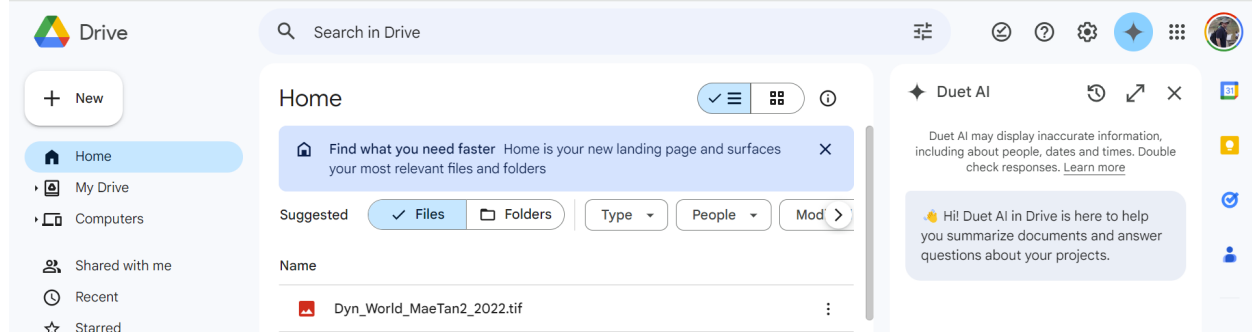
Parker Group Tutorial

January 1, 2024

//When the process is complete, you'll see a checkmark instead of spinning wheel



// NOW, go into your Google Drive and look for the file you've downloaded



// You can download this to your computer or to a hard drive, and then load it as a raster type file in
// QGIS.

// One interesting thing to do with these types of data is to generate a Zonal Histogram (which is a
// way of summarizing the land types in a defined polygon). You could even do Zonal Histograms
// for different points in time for the same area, for example to show growth of 'built' areas over
// time..

/// Want to do the same with Sentinel 2 data??! Use the following script and go through the same
// process. You don't need to select a new geographic area unless you want to.
// Remember to change your geographic area name in the JavaScript

```
// This is the Sentinel 2 collection (all the possible available Sentinel-2 imagery)
var S2_collection = ee.ImageCollection("COPERNICUS/S2")
    .filterBounds(MaeTan2)
    .filterDate('2019-01-01', '2019-02-28'); // change date range here

// This tells us what images are inside the collection
print(S2_collection);

// These are the bands that we want to be displayed
var S2_bands = ['B4', 'B3', 'B2'];

// This turns the whole S2 collection into one image, finding the middle value for each pixel
var S2_mosaic = S2_collection.median().select(S2_bands).clip(MaeTan2);

// This controls how we want the S2 image to be displayed
var S2_display = {bands: S2_bands, min: 0, max: 3000};

// This adds the S2_mosaic to the map, using the S2_display visual parameters, and giving it the
name "S2_Image"
Map.addLayer(S2_mosaic, S2_display, "S2_Image");

// This automatically pans the map to the middle of our area of interest
Map.centerObject(MaeTan2);

// This exports our Sentinel-2 image to Google Drive where we can download it
Export.image.toDrive({
  image: S2_mosaic,
  description: 'Sentinel-2',
  scale: 10,
  maxPixels: 1e13,
  region: MaeTan2
});
```