Climate Engine Virtual Workshop #2

May 17, 2023

Supporting South Florida Water Management District

Google Cloud

Climate Engine®

Contents

1. Workshop Agenda
2. Purpose
3. Instructors Introduction
4. Workshop Materials

1. Workshop Agenda

ClimateEngine®

Climate Engine Workshop Agenda

May 17, 2023

Time (ET)	**Wednesday, May 17**	**Section Lead**
Introduction (1:00-1:05)	Workshop Introduction + Overview	Caleb White
Session 1 (1:05-1:50)	API Introduction and Examples	Ankur Shah
Break (1:50-2:00)	Break	
Session 2 (2:00-2:30)	Python Script Workflows Overview	Ankur Shah / Keiko Nomura
Break (2:30-3:40)	Break	
Session 3 (2:40-3:45)	Demo Notebooks Walthrough	Ankur Shah
Wrap Up (3:45-4:00)	Wrap Up	Caleb White

Workshops Overview

Workshop #1: January 2023

This 3-hour workshop will focus on level 1 remote sensing topics, such as an introduction to remote sensing and remote sensing of algal blooms. It will then overview the example visualization app created by Climate Engine, walking through the datasets and functionalities utilized. This session will specific to algal bloom use case(s), however there will be elements of all tracks/levels of the Climate Engine Training Hub (following slides).

Workshop #2: May 2023

This 3-hour workshop will focus on on level 1 data science topics, with a focus on Climate Engine APIs. It will dive into background on APIs and resources available (documentation, swagger page, etc). It will then focus on walking users through an example workflow utilizing Google Colab notebooks. Users are encouraged to come prepared with use cases, questions and/or applications where remote sensing could be employed in their operating environment.

Workshop #3: June 2023

This 3-hour workshop will focus on level 2 & 3 remote sensing topics that will enable attendees to explore using different remotely sensed data and derived indices for various use cases. This will include a deeper dive into the example visualization apps and hosting your own creations. Users are encouraged to come prepared with use cases, questions and/or applications where remote sensing could be employed in their operating environment.

Workshop #4: July 2023

This 3-hour workshop will focus on level 2 & 3 data science topics that will enable attendees to streamline their workflows and leverage the power of cloud computing for their analysis, modeling, and visualization efforts. Users are encouraged to come prepared with use cases, questions and/or applications where remote sensing could be employed in their operating environment.

2. Purpose

Purpose

This Climate Engine virtual workshop will enable users at the South Florida Water Management District to better understand how to access data through the Climate Engine API and potential ways integrate this data into their workflows and decision making.

We will showcase the power of bringing different types of datasets together via advanced cloud pipelines, with a focus on access to the Climate Engine API through the Swagger Page and Python Scripts. We will break down the structure of API requests and how users can utilize endpoints to suit their use cases.

3. Instructor Introduction

Keiko Nomura, PhD

Director of Scientific Applications

With over 16 years of experience in remote sensing, earth observation, and policy analysis, her work has spanned various fields such as carbon market modelling in North America, and nature-based solution assessment globally. Currently serving as the Director of Science Applications at Climate Engine, Keiko leads the development of innovative remote sensing techniques that utilize big data sourced from various satellites, including NASA, ESA, and JAXA, among others, for monitoring wildfire risks, drought, and biodiversity indicators. She is also recognized internationally for her expertise in writing JavaScript for analyzing satellite data using Google Earth Engine and has widely followed online tutorials.

Ankur Shah

Geospatial Data Scientist

Ankur is a Geospatial Data Scientist at Climate Engine who assists the team with product development and environmental research. Before joining Climate Engine, he worked with a team of NASA on Earth observation projects such as detecting floating marine debris and identifying aircraft contrails using satellite imagery. Ankur holds a BSc in physics and Earth science from the University of Alabama in Huntsville. He frequently produces informative videos on environmental topics on his YouTube channel. He is passionate about natural climate solutions, sustainable urban design, and environmental education.

4. Workshop Materials

Section 1: Introduction to APIs (~45 min)

Intro to API

A - Application

P - Programming

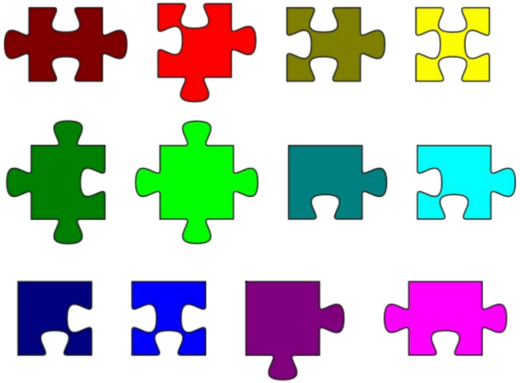
I - Interface

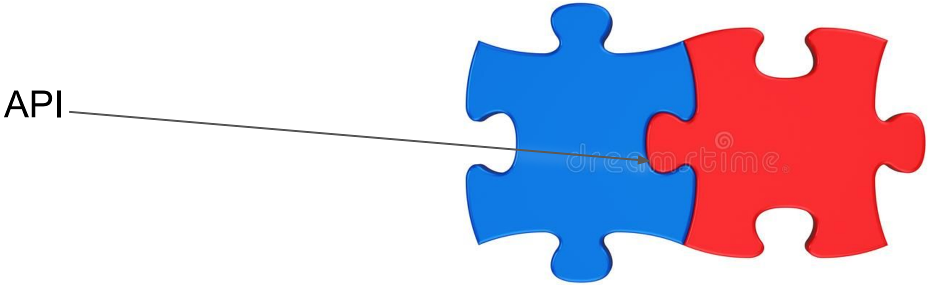
- An API is a collection of communication protocols and subroutines used by various programs to communicate between them.
- Helps two programs or applications to communicate with each other by providing them with necessary tools and functions.
- It takes the request from the user and sends it to the service provider and then again sends the result generated from the service provider to the desired user.

Intro to API

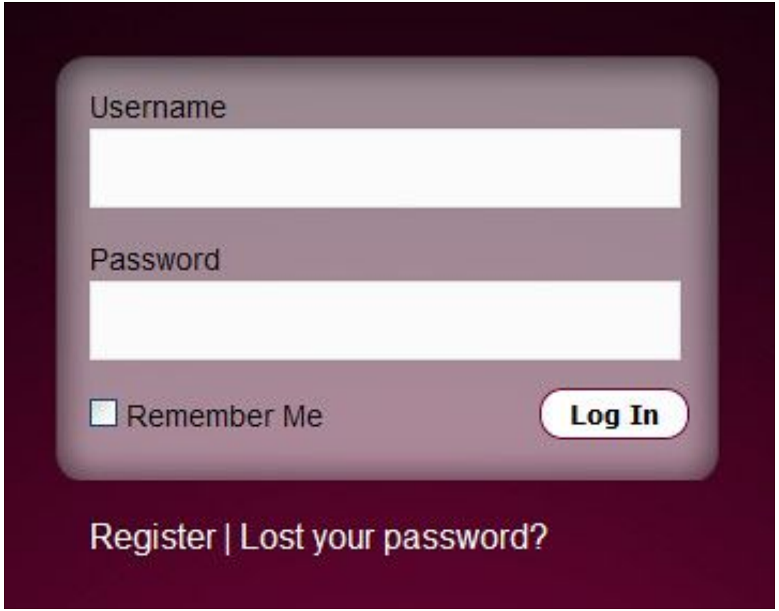
- Each puzzle piece is a separate application.

- The boundary where two pieces join is an API which facilitates communication.

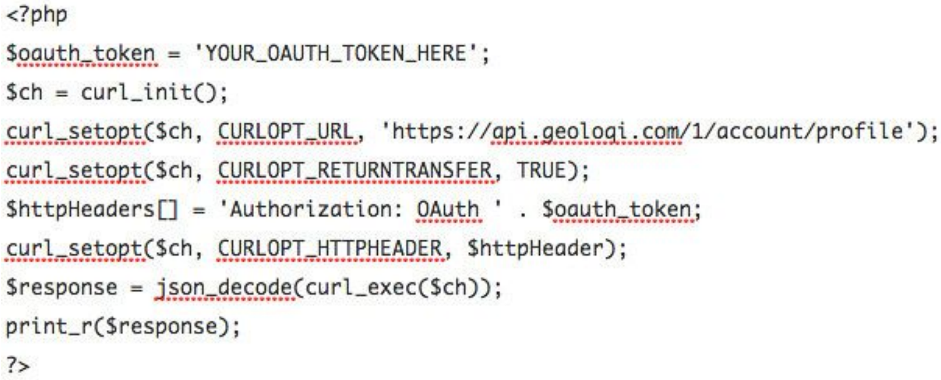




UI and API



User Interface for Users

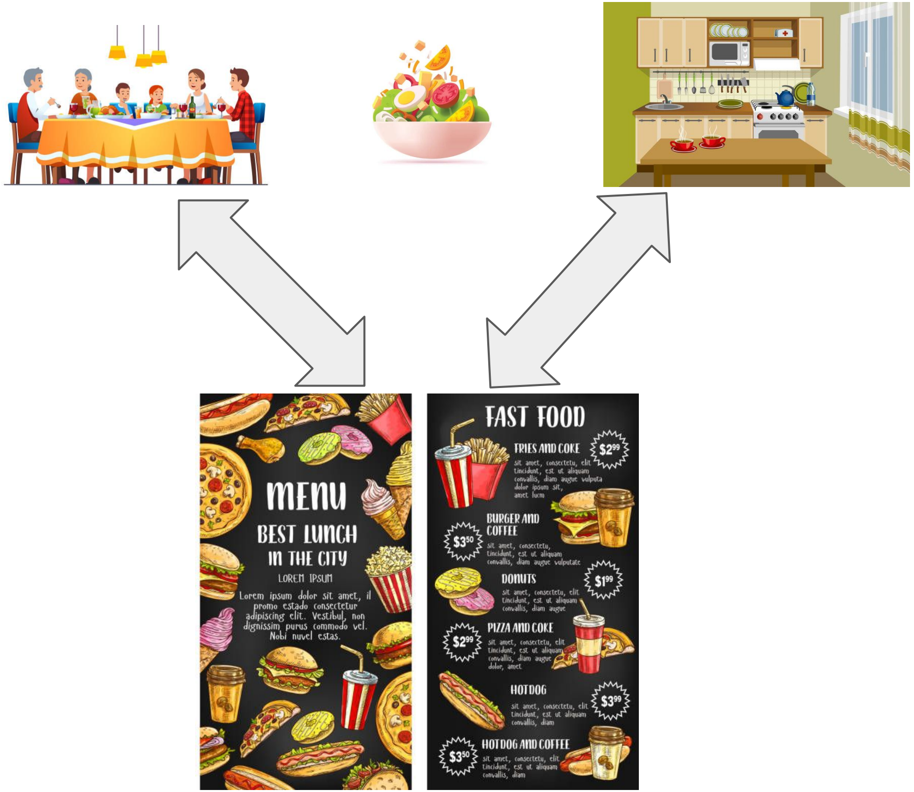


```
<?php
$oauth_token = 'YOUR_OAUTH_TOKEN_HERE';
$ch = curl_init();
curl_setopt($ch, CURLOPT_URL, 'https://api.geoloqi.com/1/account/profile');
curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);
$httpHeaders[] = 'Authorization: OAuth ' . $oauth_token;
curl_setopt($ch, CURLOPT_HTTPHEADER, $httpHeader);
$response = json_decode(curl_exec($ch));
print_r($response);
?>
```

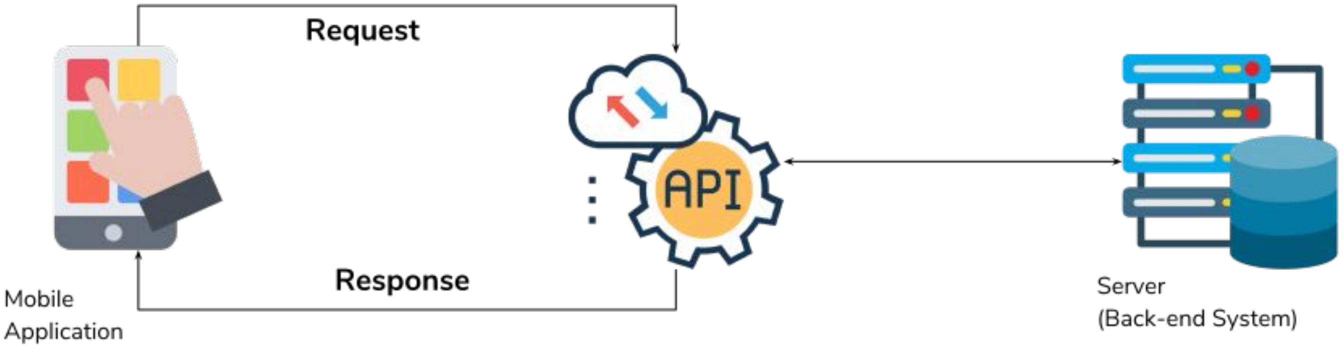
API for Code/Developers

API Analogy

- You (a client) make a request for a specific dish (data).
- The waiter (API - Application Programming Interface) takes your request and brings it to the kitchen (the server).
- The chef (the server) prepares your dish (data) and gives it to the waiter (API).
- The waiter (API) brings the dish (data) back to you (the client).
- Menu is the API documentation.
- To receive what you want (food) from the other program (restaurant), you need to make a request (place an order) in a specific way (defined by the menu).



Source: How-to Geek

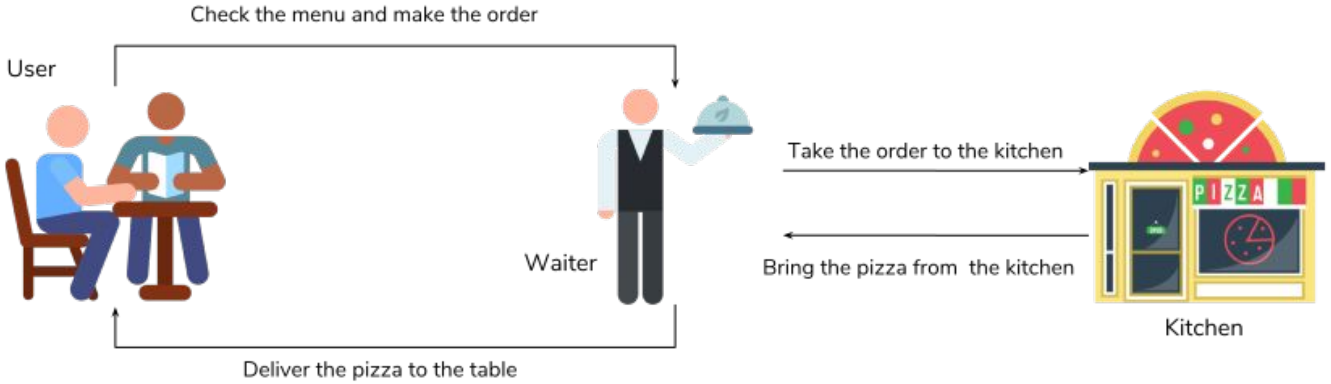


Request

Response

Mobile Application

Server
(Back-end System)



Check the menu and make the order

User

Waiter

Take the order to the kitchen

Bring the pizza from the kitchen

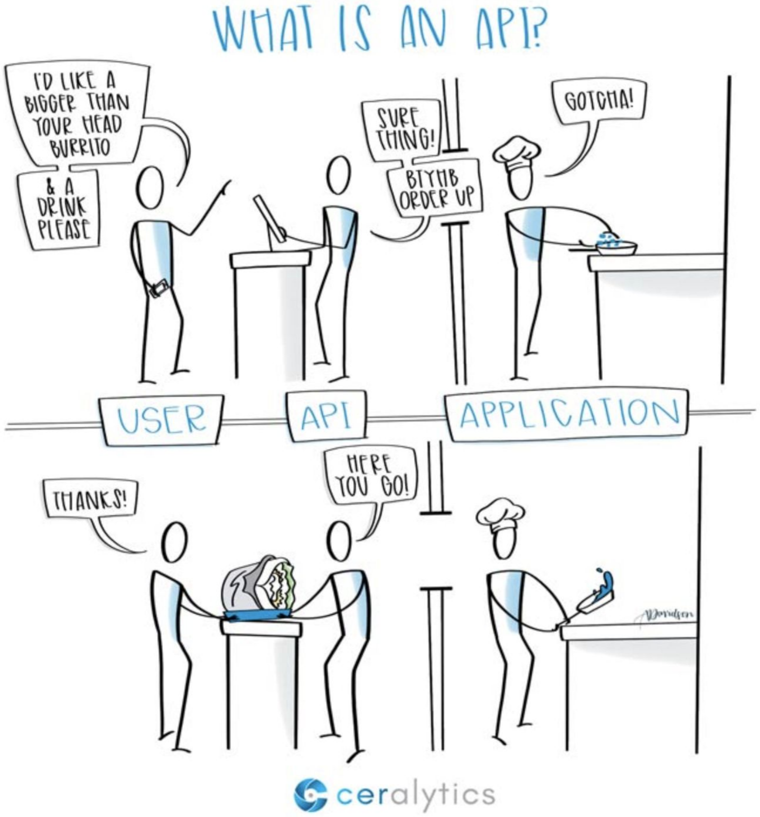
Kitchen

Deliver the pizza to the table

Source: Amanda Kothawala

What is an API?

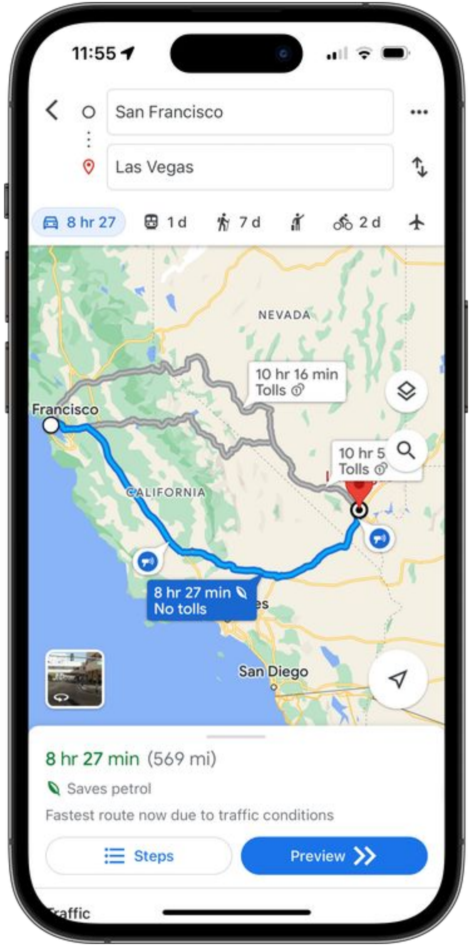
API stands for **application programming interface**. An API allows different types of softwares to communicate with each other. Using an API is critical for connecting systems that are **otherwise unable to interact with each other**.



Source: Ceralytics

Real Life API Example

- A simple example of an API is giving voice commands to your phone(i.e through Siri) to find directions to a restaurant through Google maps.
- The Siri virtual assistant and Google maps are owned by two different companies and have two completely different development teams that use different coding languages.
- An API allows Siri to send commands to Google maps and pull up the results for the user.



San Francisco

Las Vegas

8 hr 27 min (569 mi)

Saves petrol

Fastest route now due to traffic conditions

Types of API

01 **Web API**

- A web API is an application programming interface for either a web server or a web browser.
- A web API uses HTTP (Hyper Text Transfer Protocol).

02 **Local API**

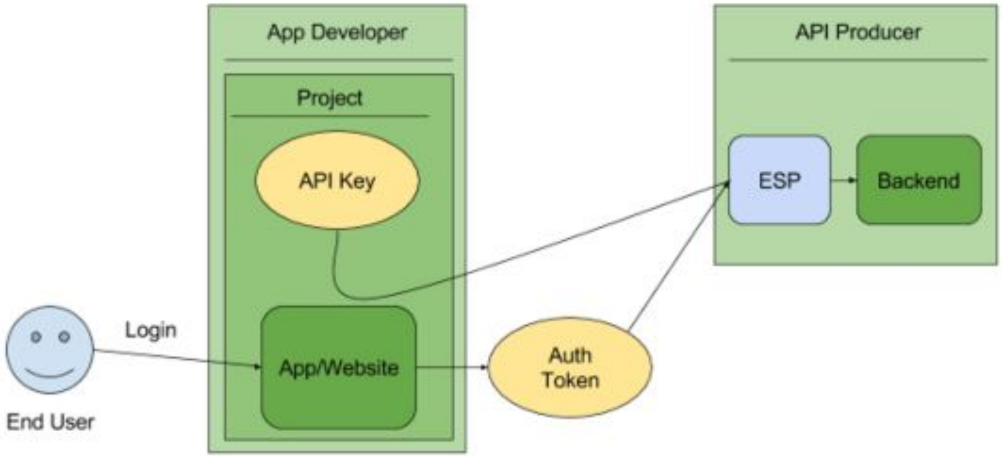
- A local API enables communication between software programs or applications running on the same device or network.
- Local APIs are commonly used in mobile apps, desktop software, and IoT devices.

03 **Program API**

- A program API is an interface designed to enable communication between software programs or applications running on different devices or networks.

What is an API Key and Why it Matters?

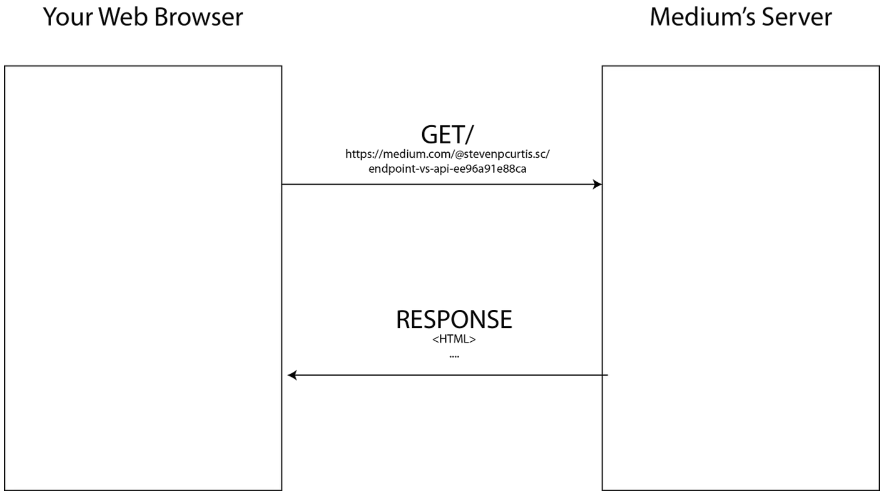
- API keys identify the calling project i.e the application or site — making the call to an API.
- Authentication tokens identify a user aka the person — that is using the app or site.
- Project and user authorization is critical for API access



Source: Google Cloud

What is an Endpoint?

- **Endpoints** are critical to understand in the larger API interface. An endpoint is **one end in the communication channel between the two applications**. An API Endpoint is the **URL** of a server or service.
- An application will use an endpoint to process a function with either a ‘response’ or a ‘request.’ To correctly send a request or interpret a response, a **strict protocol** established by the API must be followed.



Your Web Browser

Medium’s Server

GET/
https://medium.com/@stevenpcurtis.sc/endpoint-vs-api-ee96a91e88ca

RESPONSE
<HTML>
....

Endpoint Analogy - Toy Box

- Imagine you have a giant toy box filled with all kinds of toys: cars, dolls, puzzles, building blocks, and so on.

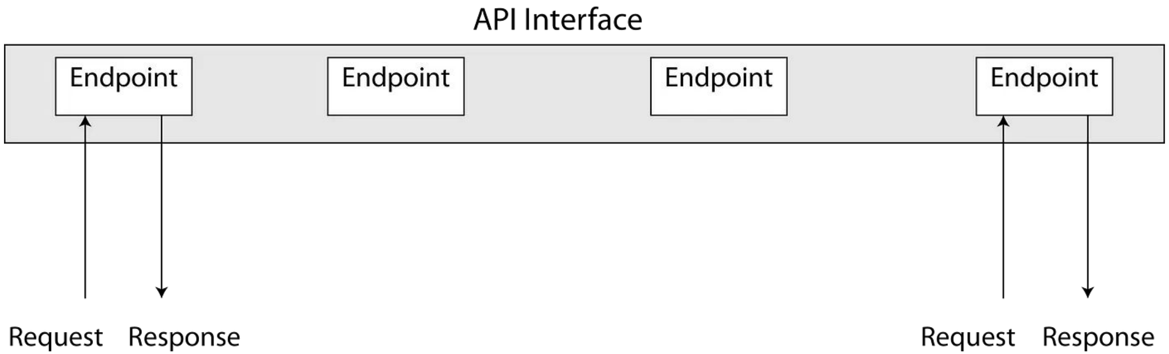
- Each type of toy in that box can be considered an "endpoint". So if you want to play with cars, you ask the "car endpoint" in your toy box, and you get a car, etc.

- So, in the world of APIs, when a computer program wants a specific type of data, it asks a specific "endpoint" - just like you would ask for a specific type of toy from your toy box. In this way, each type of toy in your toy box is like an API endpoint. It's a specific place you can ask for a specific thing you want.

Endpoint examples

- The Twitter API exposes data about tweets, direct messages, users, and more.
- Let’s say you want to retrieve the content of a specific tweet. To do this, you can use the tweet lookup endpoint, which has the URL

  https://api.twitter.com/2/tweets/{id} (where {id} is the unique identifier of the tweet).



How to use a Web API

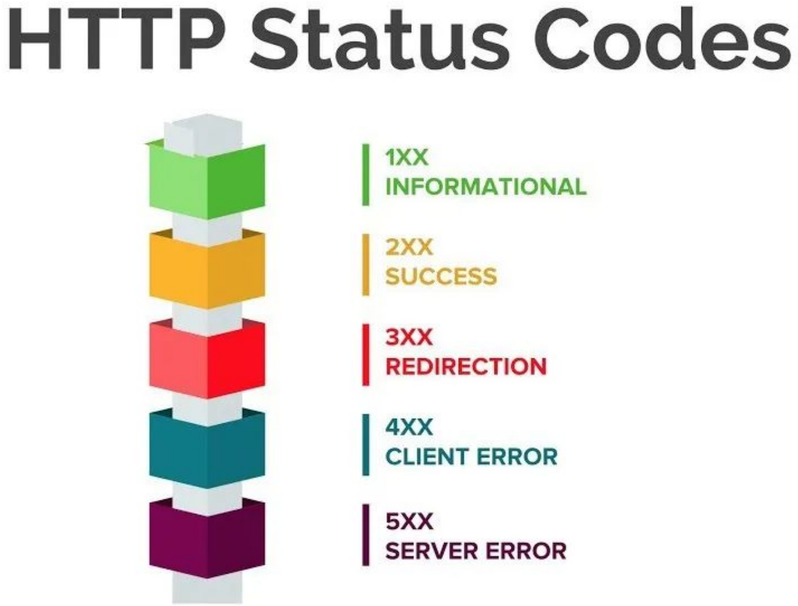
When we interact with Web APIs, the four most common

HTTP methods in API requests are:

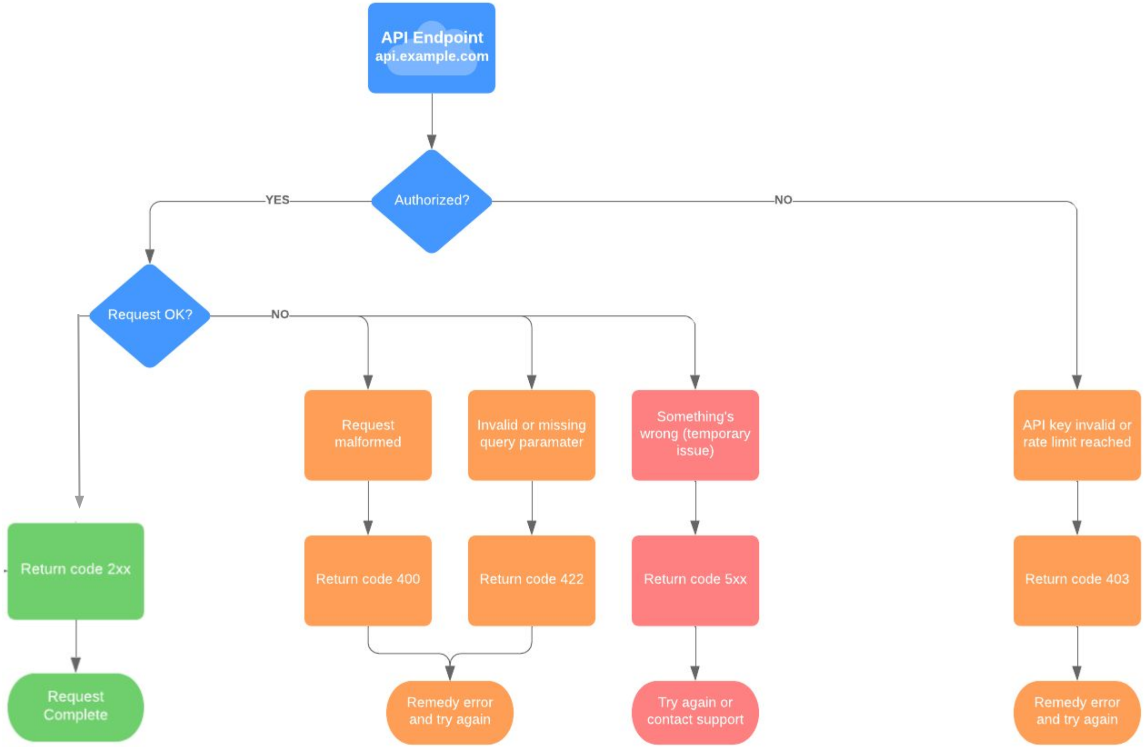
- **GET**: retrieves a resource
- **POST**: creates a resource
- **PUT**: updates an existing resource
- **DELETE**: removes a resource

Status Codes

- ***200*** - The request is **successful** as the endpoint does exist and makes some internal validation, but the response has to include some additional information on access.
- ***401* -** User has invalid credentials**.** No extra information required on the response.
- ***403* -** Valid credentials but the access is **forbidden** and the response includes information on access denied.
- ***503* -** Cannot do anything unless user login is validated and authorized (**service unavailable**).



Source: Jose 1 Santa Cruz, https://www.youtube.com/watch?v=LtNSd_4txVc



API Endpoint
api.example.com

Authorized?

YES

NO

Request OK?

NO

Return code 2xx

Request Complete

Request malformed

Invalid or missing query paramater

Something's wrong (temporary issue)

API key invalid or rate limit reached

Return code 400

Return code 422

Return code 5xx

Return code 403

Remedy error and try again

Try again or contact support

Remedy error and try again

Source: Lucidchart

How APIs can be leveraged

- Gather, process, analyze, and visualize data in the cloud without having to visit multiple websites and download large amounts of data.

- Save time + effort in Earth Engine writing and executing scripts to format data for analysis.

- Automate analysis, i.e. export time series for multiple areas of interest by utilizing for loops.

Climate Engine API Endpoint Types

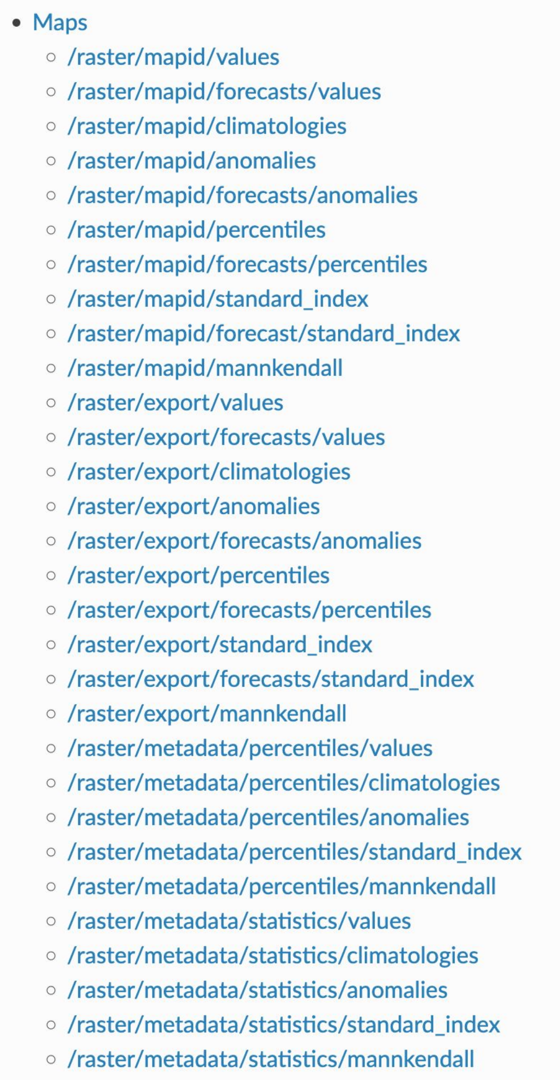
Maps	Time Series
Zonal Statistics	Probabilities

Maps Endpoints

1) **Mapid** - these endpoints are used to generate Earth Engine maps, and produce an Earth Engine map ID value that you can visualize.

2) **Export** - there endpoints are used to generate a map and export it to a Google Cloud Storage bucket.

3) **Metadata** - these endpoints return percentiles and statistics of map values.



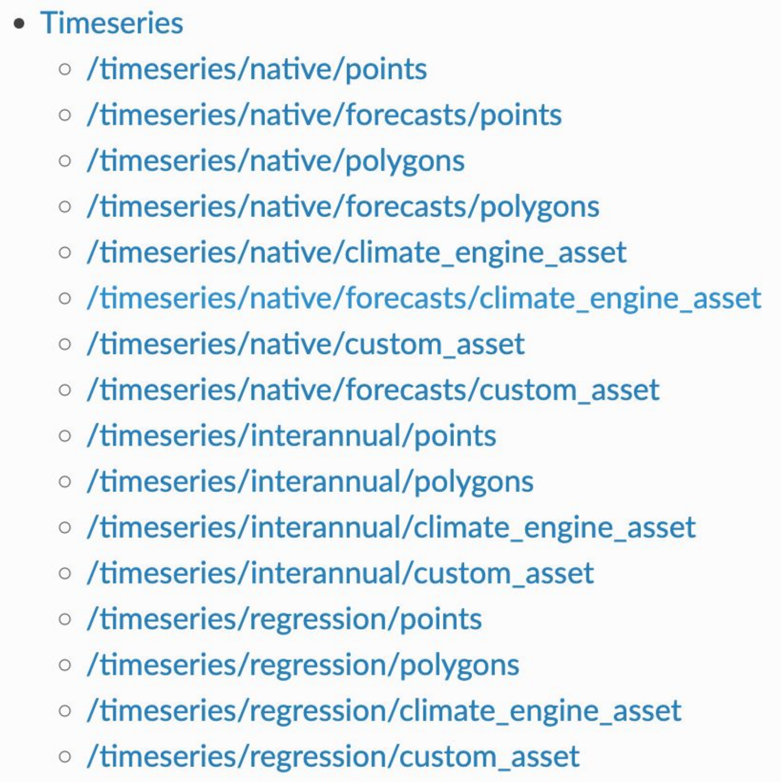
- Maps
  - /raster/mapid/values
  - /raster/mapid/forecasts/values
  - /raster/mapid/climatologies
  - /raster/mapid/anomalies
  - /raster/mapid/forecasts/anomalies
  - /raster/mapid/percentiles
  - /raster/mapid/forecasts/percentiles
  - /raster/mapid/standard_index
  - /raster/mapid/forecast/standard_index
  - /raster/mapid/mannkendall
  - /raster/export/values
  - /raster/export/forecasts/values
  - /raster/export/climatologies
  - /raster/export/anomalies
  - /raster/export/forecasts/anomalies
  - /raster/export/percentiles
  - /raster/export/forecasts/percentiles
  - /raster/export/standard_index
  - /raster/export/forecasts/standard_index
  - /raster/export/mannkendall
  - /raster/metadata/percentiles/values
  - /raster/metadata/percentiles/climatologies
  - /raster/metadata/percentiles/anomalies
  - /raster/metadata/percentiles/standard_index
  - /raster/metadata/percentiles/mannkendall
  - /raster/metadata/statistics/values
  - /raster/metadata/statistics/climatologies
  - /raster/metadata/statistics/anomalies
  - /raster/metadata/statistics/standard_index
  - /raster/metadata/statistics/mannkendall

Timeseries Endpoints

1) **Native** - these endpoints are used to generate ‘raw’ time series data over points/polygons/features for Climate Engine datasets.

2) **Interannual** - endpoints are used to generate a time series of yearly values of the dataset variable summarized over a season.

3) **Regression** - these endpoints are used to perform regression analysis of the dataset variable summarized over a season.



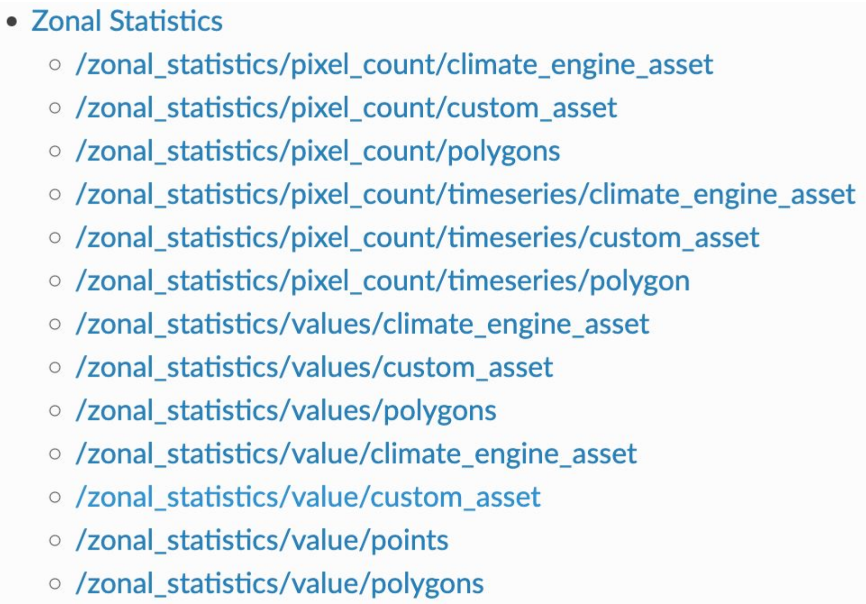
- Timeseries
  - /timeseries/native/points
  - /timeseries/native/forecasts/points
  - /timeseries/native/polygons
  - /timeseries/native/forecasts/polygons
  - /timeseries/native/climate_engine_asset
  - /timeseries/native/forecasts/climate_engine_asset
  - /timeseries/native/custom_asset
  - /timeseries/native/forecasts/custom_asset
  - /timeseries/interannual/points
  - /timeseries/interannual/polygons
  - /timeseries/interannual/climate_engine_asset
  - /timeseries/interannual/custom_asset
  - /timeseries/regression/points
  - /timeseries/regression/polygons
  - /timeseries/regression/climate_engine_asset
  - /timeseries/regression/custom_asset

Zonal Statistics Endpoints

1) **Pixel Count** - these endpoints produce statistics and time series statistics on the number of pixels in each raster category for polygons and custom assets.

2) **Values** - these endpoints compute temporal statistics which are reduced over specified geometric features.

3) **Value** - these endpoints return values for statistic datasets (no time dependence) over different specified geometries.



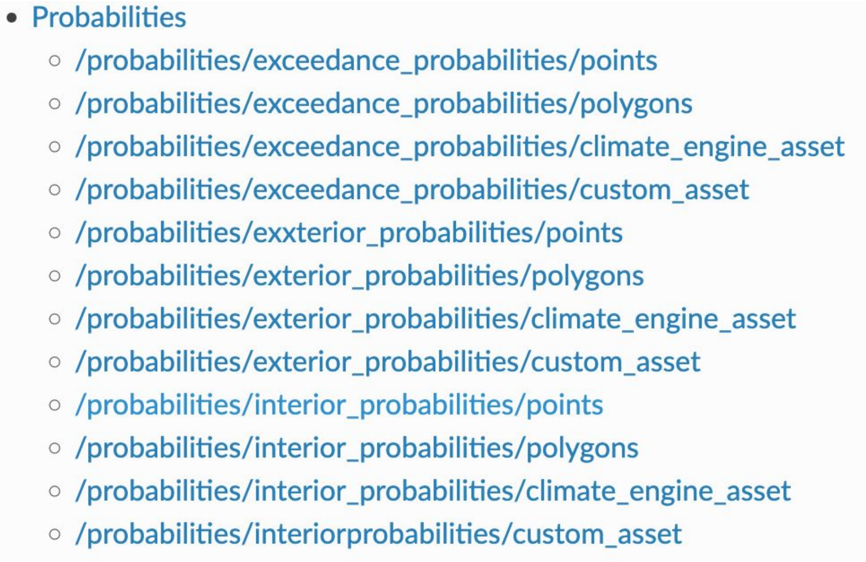
- Zonal Statistics
  - /zonal_statistics/pixel_count/climate_engine_asset
  - /zonal_statistics/pixel_count/custom_asset
  - /zonal_statistics/pixel_count/polygons
  - /zonal_statistics/pixel_count/timeseries/climate_engine_asset
  - /zonal_statistics/pixel_count/timeseries/custom_asset
  - /zonal_statistics/pixel_count/timeseries/polygon
  - /zonal_statistics/values/climate_engine_asset
  - /zonal_statistics/values/custom_asset
  - /zonal_statistics/values/polygons
  - /zonal_statistics/value/climate_engine_asset
  - /zonal_statistics/value/custom_asset
  - /zonal_statistics/value/points
  - /zonal_statistics/value/polygons

Probabilities Endpoints

1) **Exceedance Probabilities** - these endpoints produce the probability of a variable value to exceed a threshold at a location defined by a geometry.

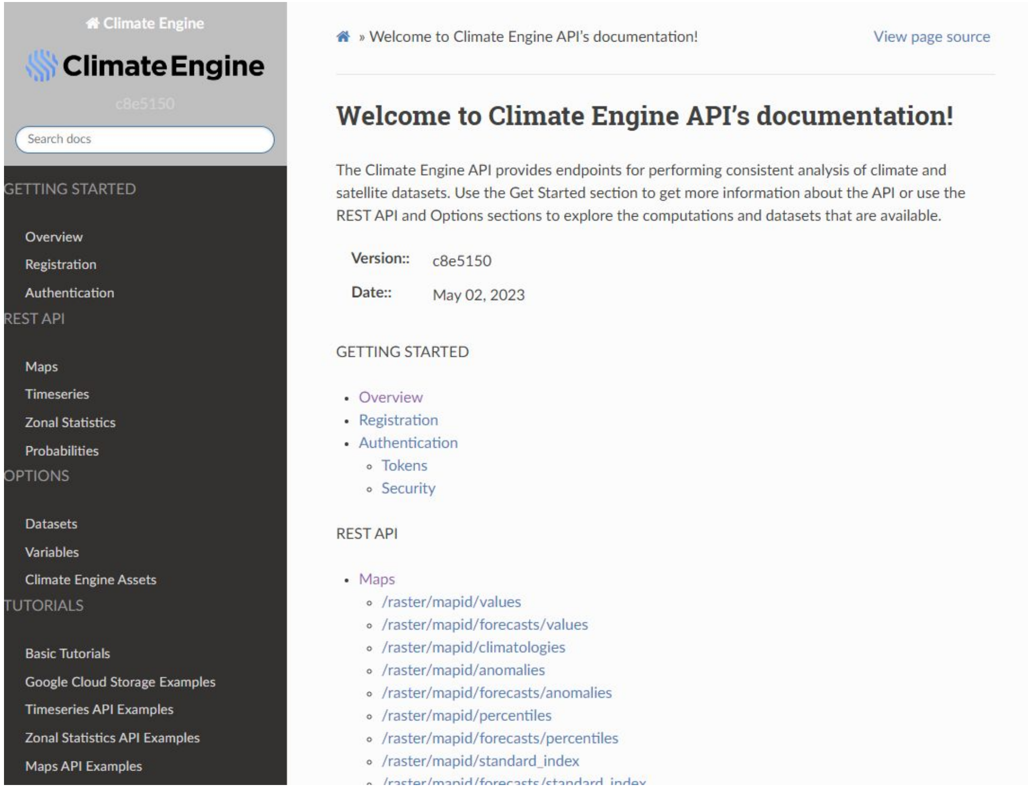
2) **Exterior Probabilities** - these endpoints produce the probability of a variable value to lie outside of a lower and an upper threshold at a location defined by a geometry.

3) **Interior Probabilities** - these endpoints produce the probability of a variable value to lie inside of a lower and an upper threshold at a location defined by a geometry.



- Probabilities
  - /probabilities/exceedance_probabilities/points
  - /probabilities/exceedance_probabilities/polygons
  - /probabilities/exceedance_probabilities/climate_engine_asset
  - /probabilities/exceedance_probabilities/custom_asset
  - /probabilities/exxterior_probabilities/points
  - /probabilities/exterior_probabilities/polygons
  - /probabilities/exterior_probabilities/climate_engine_asset
  - /probabilities/exterior_probabilities/custom_asset
  - /probabilities/interior_probabilities/points
  - /probabilities/interior_probabilities/polygons
  - /probabilities/interior_probabilities/climate_engine_asset
  - /probabilities/interiorprobabilities/custom_asset

Climate Engine API Documentation - Link



Welcome to Climate Engine API's documentation!

The Climate Engine API provides endpoints for performing consistent analysis of climate and satellite datasets. Use the Get Started section to get more information about the API or use the REST API and Options sections to explore the computations and datasets that are available.

Version:: c8e5150

Date:: May 02, 2023

GETTING STARTED

- Overview
- Registration
- Authentication
  - Tokens
  - Security

REST API

- Maps
  - /raster/mapid/values
  - /raster/mapid/forecasts/values
  - /raster/mapid/climatologies
  - /raster/mapid/anomalies
  - /raster/mapid/forecasts/anomalies
  - /raster/mapid/percentiles
  - /raster/mapid/forecasts/percentiles
  - /raster/mapid/standard_index
  - /raster/mapid/forecasts/standard_index

Interactive Session

Q & A

-

Section 2: Python Script Workflows Overview (~20 min)

Popular Geospatial Python Packages

- **Numpy** - helps users work with numbers, arrays (including images), and do math more easily
- **Geopandas** -facilitates work with maps and geographical data
- **Matplotlib** - helps users make graphs, charts, and maps
- **Seaborn** - helps users create more advance and fancy-looking graphs and charts
- **Folium** - helps create interactive maps inside Jupyter notebook or Colab
- **Gdal** - extremely useful for working with different types of geospatial data
- **Rasterio** - helps people work with raster data creation and processing

Making API requests through Python Scripts

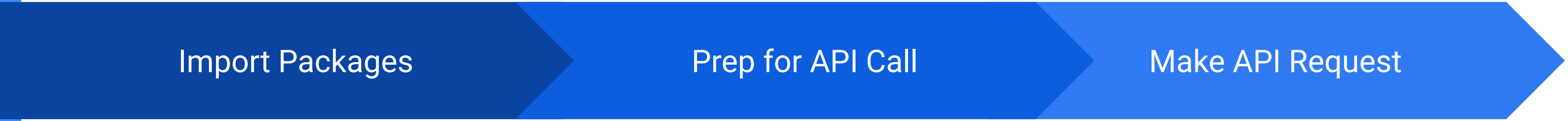
Utilizing python scripts to execute Climate Engine API requests allows you to customize your analysis workflow and automate analyses.

Setting up a python script for requests can be broken down into a few steps:

1. Identify an endpoint of interest and define it as a variable in your script
2. Look up the parameter requirements and options on the docs page (ensure you use proper parameter names for datasets and variables)
3. Input the path for your google cloud storage bucket to store exports

   Note: Some endpoints don't export to a bucket, but are returned as json and can be exported utilizing functions in script
4. Send request to the API
5. Track status updates as request is processed

Making Raster Requests & Exporting to Google Cloud Storage



Import Packages

- Import packages for
  - Making HTTP requests
  - Working with raster data
  - Making Maps

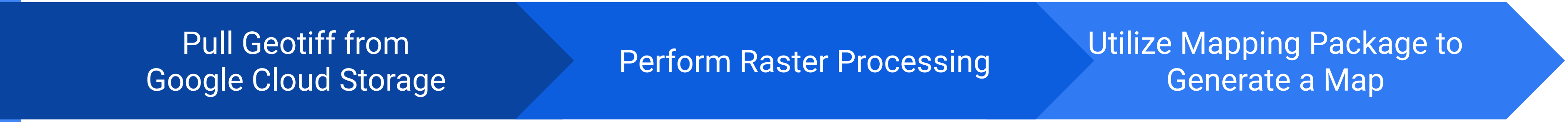
Prep for API Call

- Select an endpoint
- Define parameters
- Define output path to google cloud storage

Make API Request

- Send request to the API
- Print response to see task status

Copying Raster to Notebook & Generating Maps



Pull Geotiff from Google Cloud Storage

- Authenticate access to google cloud
- Copy file from google cloud storage bucket to local content bucket

Perform Raster Processing

- Open geotiff with raster package such as gdal or rasterio

Utilize Mapping Package to Generate a Map

- Create a figure to add the raster to
- Add a color bar, set color ramp, and set min/max
- Add title
- Add option to export as png
- Show map to see resulting map

Making Time Series Requests

Import Packages

- Import packages for
  - Making HTTP requests
  - Working with tabular data
  - Making Charts

Prep + Make API Call

- Select endpoint of interest
- Define parameters
- Make request

Turn Response into Dataframe

- Select data dictionary from unpacked response
- Use pd.DataFrame.from_dict to convert dictionary into pandas dataframe

Creating Graphs in Notebooks

Process Results for Plotting

- Get rid of non-valid values
- Format date for plotting
- Set x values to date variable
- Set y values to value variable

Create Plot

- Create Figure
- Plot date and value
- Add labels and title
- Show plot to check results

Export PNGs

- Export png + download

Questions?

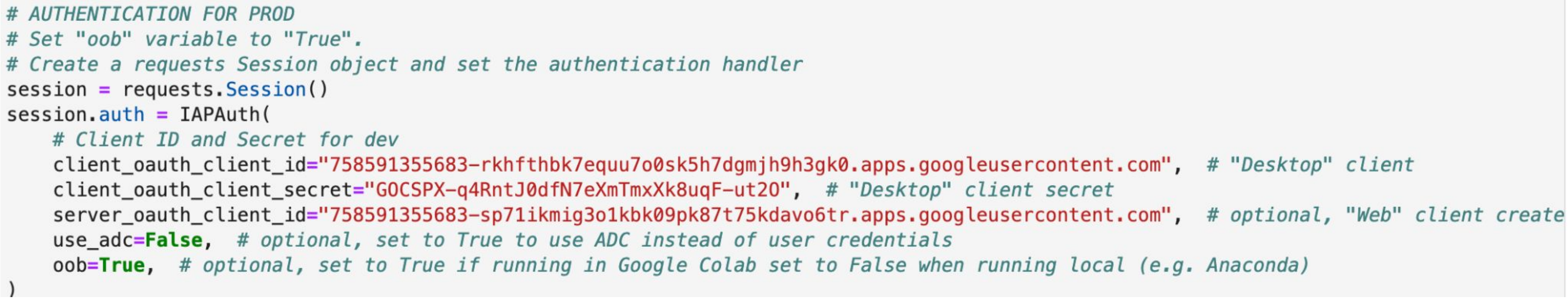
Q & A

-

5 minute Break

Section 3: API Notebook Walkthrough (~75 min)

This slide will be used to show that Climate Engine will be showing this in our tenant, but SFWMD will be using their own auth & API urls



```
# AUTHENTICATION FOR PROD
# Set "oob" variable to "True".
# Create a requests Session object and set the authentication handler
session = requests.Session()
session.auth = IAPAuth(
    # Client ID and Secret for dev
    client_oauth_client_id="758591355683-rkhfthbk7equu7o0sk5h7dgmjh9h3gk0.apps.googleusercontent.com",  # "Desktop" client
    client_oauth_client_secret="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",  # "Desktop" client secret
    server_oauth_client_id="758591355683-sp71ikmig3o1kbk09pk87t75kdavo6tr.apps.googleusercontent.com",  # optional, "Web" client create
    use_adc=False,  # optional, set to True to use ADC instead of user credentials
    oob=True,  # optional, set to True if running in Google Colab set to False when running local (e.g. Anaconda)
)
```

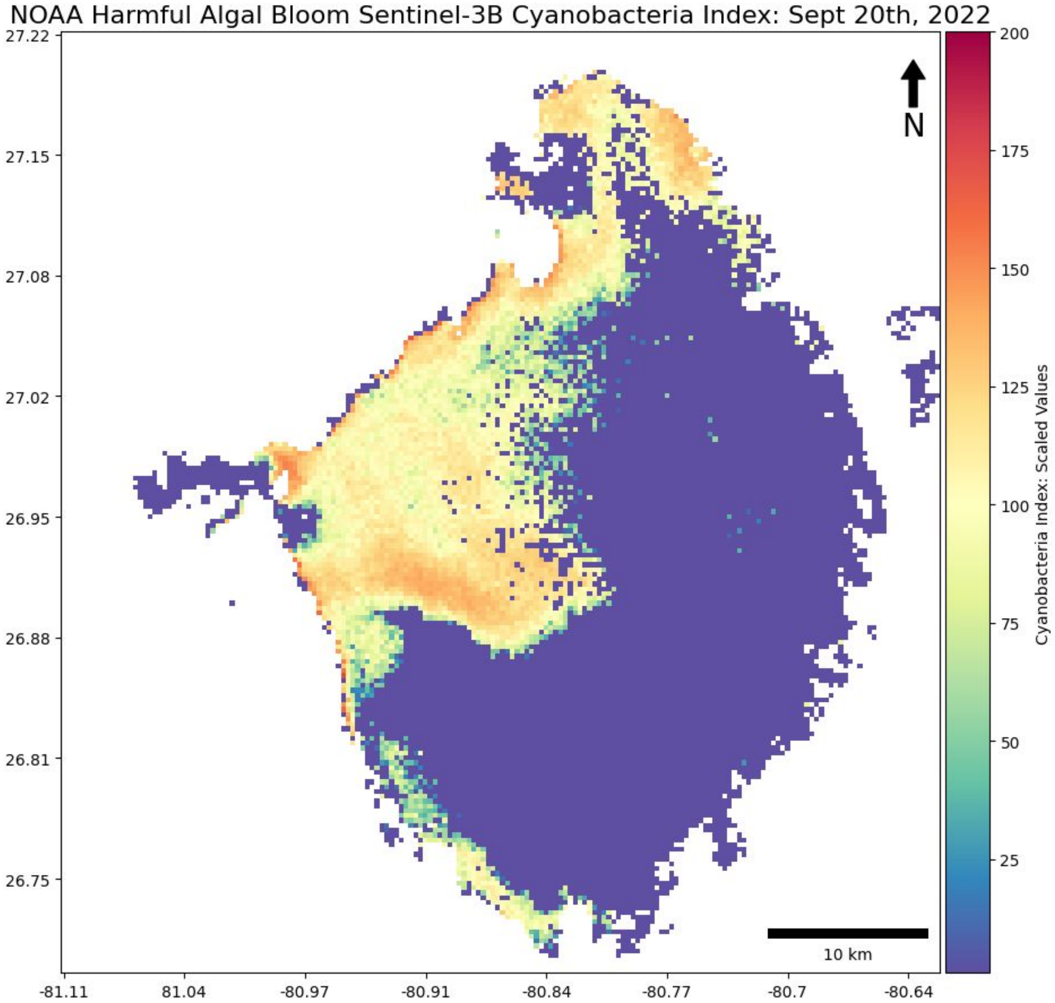
Raster Data Exports (NOAA HAB)

The National Oceanic and Atmospheric Administration (NOAA) Harmful Algal Bloom (HAB) data is a crucial resource for monitoring and assessing toxic algae outbreaks in coastal waters. These blooms can have severe consequences for marine life, human health, and local economies. NOAA's HAB data encompasses a range of information, including satellite imagery, field observations, and model forecasts, providing scientists and decision-makers with the tools needed to detect, predict, and mitigate the impacts of these harmful events.

1. Install Relevant Packages
2. Export NOAA HAB S3A cicyano band tif for a known date
3. Export NOAA HAB S3B cicyano band tif for a known date
4. Define raster operations to acquire coordinates and values of the geotiffs
5. Plot the cicyano index for a map

The plot on the right displays the strength of the cicyano index which is an indicator of surface algal blooms.

The colormap used is 'Spectral_r', which ranges from red (indicating high cyanobacteria index values) to purple (indicating low cyanobacteria index values).

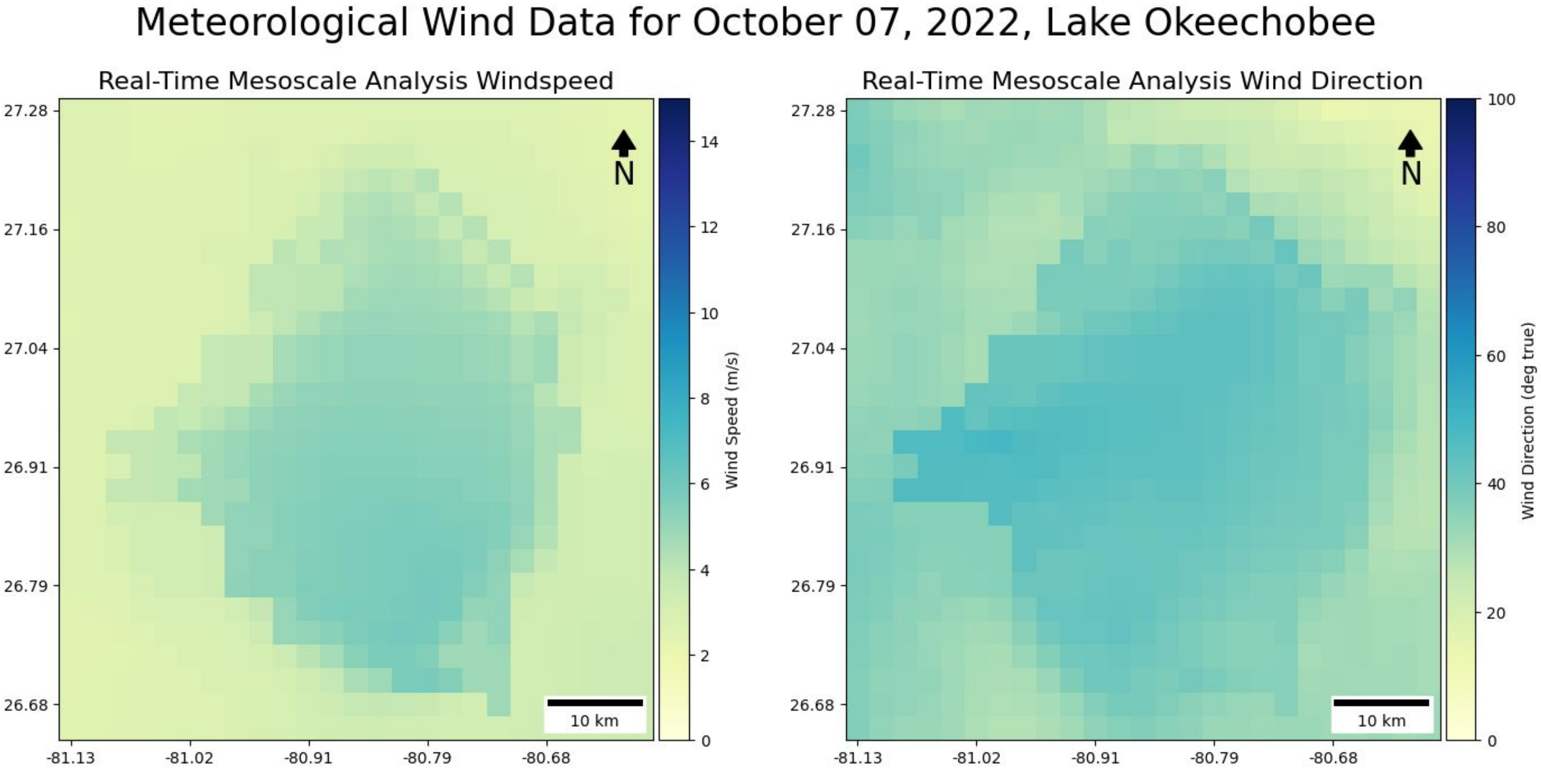


Raster Data Exports (RTMA)

The Real-Time Mesoscale Analysis (RTMA) wind data is a high-resolution, regularly updated dataset produced by the National Oceanic and Atmospheric Administration (NOAA) in the United States. The primary objective of RTMA is to provide accurate and timely surface weather analysis, including wind speed and direction, across the nation.

1. Install Relevant Packages
2. Export RTMA Wind Speed Tif
3. Export RTMA Wind Direction Tif
4. Define raster operations to acquire coordinates and values of the geotiffs
5. Plot the wind speed and wind direction side by side in a dual plot

Raster Data Exports (RTMA)


Meteorological Wind Data for October 07, 2022, Lake Okeechobee
Real-Time Mesoscale Analysis Windspeed
Real-Time Mesoscale Analysis Wind Direction
27.28
27.16
27.04
26.91
26.79
26.68
-81.13
-81.02
-80.91
-80.79
-80.68
N
10 km
Wind Speed (m/s)
0
2
4
6
8
10
12
14
Wind Direction (deg true)
0
20
40
60
80
100

Automating Time Series Exports

1. Install Relevant Packages
2. Create List of Locations to Iterate Through
3. Write Function to Request Timeseries and Export CSVs
4. Apply function to List of Locations
5. Check Google Cloud Storage for CSVs

FDEP Data Access

1. Install Relevant Packages
2. Create a list of properties within the FDEP dataset
3. Create a list of IDs as desired
4. Call API to export a csv of FDEP data for IDs
5. Check Google Cloud Storage for CSVs
6. Use the resulting CSVs to create dataframes and plot the data as needed

Thank you!

Climate Engine®

Google Cloud
Partner