



Earth Engine Makerspace

Resources for Earth Engine Developers, by Earth Engine
Developers



eemont: A python package that extends Google Earth Engine

David Montero Loaiza

Research Assistant

Remote Sensing Centre for Earth Systems Research, University of Leipzig



Geo for Good Summit 2021

Common Earth Engine code

from   to

```
def cloudMask(img):
    qa = img.select('QA60')
    cloudBitMask = 1 << 10
    cirrusBitMask = 1 << 11
    mask = qa.bitwiseAnd(cloudBitMask).eq(0)
        .And(qa.bitwiseAnd(cirrusBitMask).eq(0))
    return img.updateMask(mask)

def scale(img):
    scaling = img.select(['B.*'])
    x = scaling.multiply(0.0001)
    scaling = img.select(['AOT', 'WVP'])
    x = x.addBands(scaling.multiply(0.001))
    notScaling = img.select(['SCL', 'T.*', 'M.*', 'Q.*'])
    x = x.addBands(notScaling)
    return x

def addIndices(img):
    a = img.normalizedDifference(['B8', 'B4']).rename('NDVI')
    b = img.normalizedDifference(['B8', 'B3']).rename('GNDVI')
    c = img.normalizedDifference(['B3', 'B11']).rename('NDSI')
    return img.addBands([a,b,c])

S2 = ee.ImageCollection("COPERNICUS/S2_SR") \
    .map(cloudMask) \
    .map(scale) \
    .map(addIndices)
```

eemont styled code

```
S2 = ee.ImageCollection("COPERNICUS/S2_SR") \
    .maskClouds() \
    .scaleAndOffset() \
    .spectralIndices(["NDVI", "GNDVI", "NDSI"])
```

Raster methods

Clouds and Shadows Masking

```
S2 = ee.ImageCollection("COPERNICUS/S2_SR") \
    .maskClouds(prob = 70,buffer = 300,cldi = -0.5)
```

Scaling and Offsetting

```
L8 = ee.ImageCollection("LANDSAT/LC08/C02/T1_L2") \
    .scaleAndOffset()
```

Spectral Indices | @awesome-spectral-indices

```
S2 = ee.ImageCollection("COPERNICUS/S2_SR") \
    .scaleAndOffset() \
    .spectralIndices(["NDVI", "kNDVI", "NDWI"])
```

Panchromatic Sharpening | @aazuspan

```
L8 = ee.ImageCollection("LANDSAT/LC08/C01/T1_TOA") \
    .panSharpen(method="HPFA", qa=["MSE", "RMSE"], maxPixels=1e13)
```

Histogram Matching | @aazuspan

```
source = ee.Image("LANDSAT/LC08/C01/T1_TOA/LC08_047027_20160819")
target = ee.Image("LANDSAT/LE07/C01/T1_TOA/LE07_046027_20150701")
m = source.matchHistogram(target, {"B4":"B3", "B3":"B2", "B2":"B1"})
```



Vector methods

Time Series by Region (or Regions)

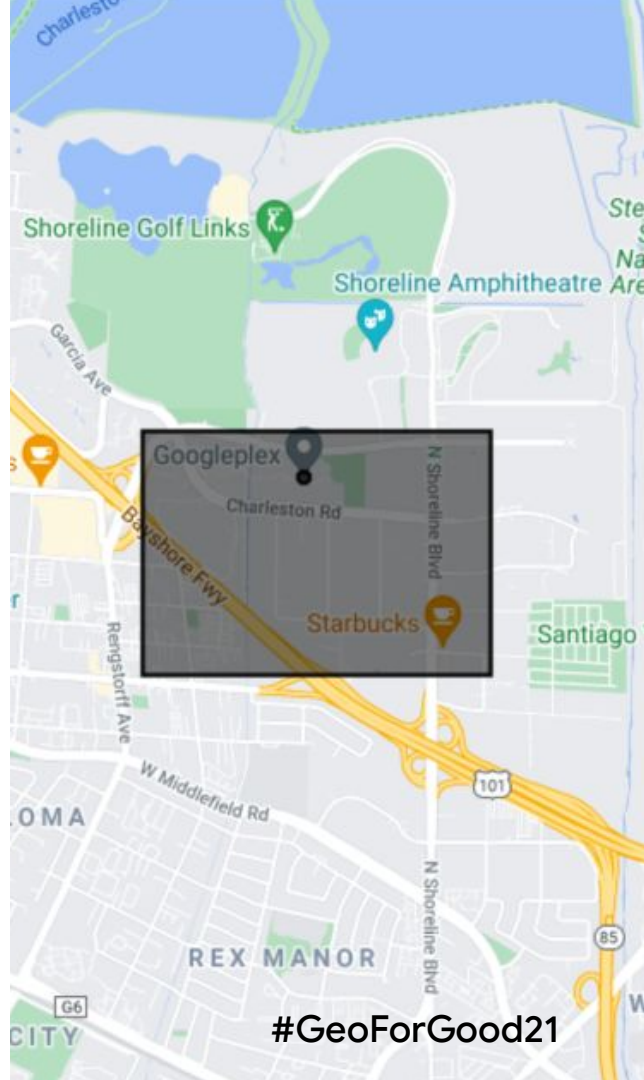
```
ts = ee.ImageCollection("COPERNICUS/S2_SR") \
    .filterBounds(fc).maskClouds().scaleAndOffset() \
    .spectralIndices(["EVI","NDVI"]) \
    .getTimeSeriesByRegions(reducer = ee.Reducer.median(),
                             collection = fc,
                             bands = ["EVI","NDVI"],
                             scale = 10)
```

Constructors by Queries | @geopy

```
ua = "GeoForGoodSummit2021-eemont"  
Gbbox = ee.Geometry.BBoxFromQuery("Googleplex",user_agent = ua)
```

Constructors by Plus Codes | @aazuspan @openlocationcode

```
pc = "CWC8+R9 Mountain View, California, USA"  
Gpc = ee.Geometry.PointFromPlusCode(pc,user_agent = ua)
```



#GeoForGood21

Operators and Containers

Overloaded Operators

```
math = ee.Image(1.0) + ee.Number(2.0) * 0.3
```

Selecting Bands

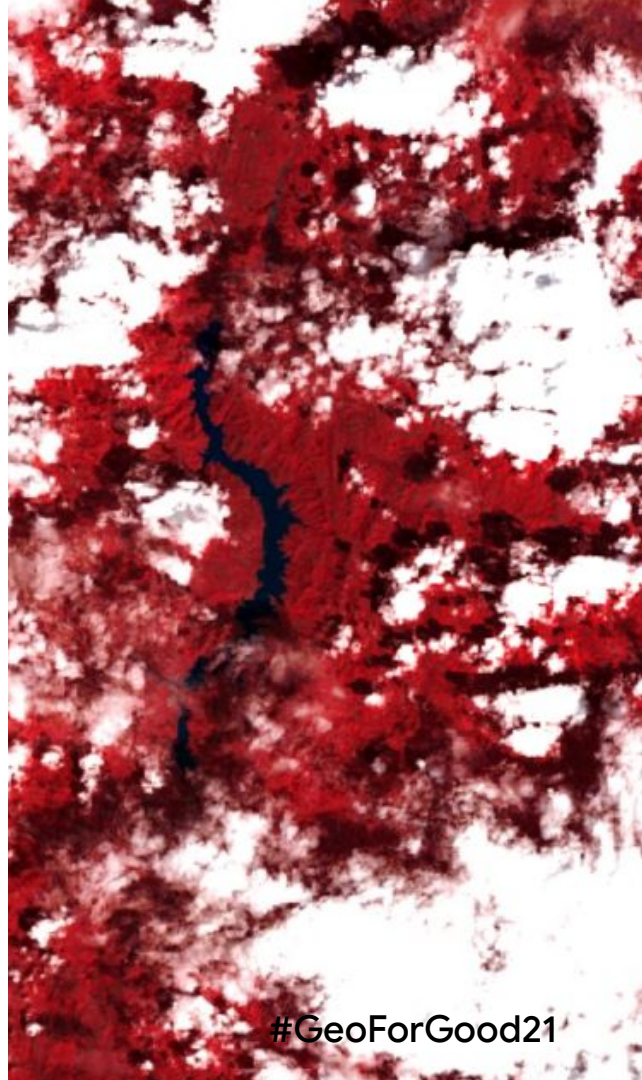
```
S2 = ee.ImageCollection("COPERNICUS/S2_SR").first()  
RGBa = S2[["B2", "B3", "B4"]]  
RGBb = S2[1:4]  
ALL = S2["B.*"]
```

Selecting Columns

```
WDPA = ee.FeatureCollection("WCMC/WDPA/current/polygons")  
WDPA = WDPA[['WDPAID', 'NAME', 'REP_AREA']]
```

Collection Length

```
len(S2)  
len(WDPA)
```



What's next?

The eeExtra Ecosystem

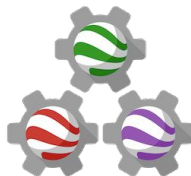
+ @csaybar (César Aybar Camacho)



**QGIS Earth Engine
Plugin** @gena et al. |
*Integrates Google
Earth Engine and QGIS
using Python API*



EarthEngine.jl
@KMarkert | *Google
Earth Engine in Julia*



eeExtra @davemlz
@csaybar | *A ninja
python package that
unifies the Google
Earth Engine
ecosystem*



eeMont @davemlz | *A
python package that
extends Google Earth
Engine*



rgeeExtra @csaybar |
*High-level functions to
process spatial and
simple Earth Engine
objects*



Resources that can be associated

Directly Associated Packages



Contributions to eemont (*and eeExtra*) are welcome!