

Python 3.11.3 [main, GCC 13.1.1 20230429]
Linux-6.4.3-arch1-2-x86_64-with-glibc2.37

[Module Index](#) : [Topics](#) : [Keywords](#)

Get

Search

depal (version 0.0.1)

[index](#)
</home/sachin/Projects/depal/depal.py>

depal.py: Digital Earth Pacific (Abstraction Library)

Modules

[cartopy.crs](#)
[geopandas](#)
[itertools](#)
[matplotlib](#)

[xrspatial.multispectral](#)
[numpy](#)
[planetary_computer](#)
[pandas](#)

[matplotlib.pyplot](#)
[pystac_client](#)
[rasterio](#)
[rioxarray](#)

[stackstac](#)
[xarray](#)

Functions

```
chart_global_land_cover(data)
    # Annual Charting of Land Cover Classes

cleanup()
    # Cleanup Dask Resources

colour_maps()
    # List Colour Maps

do_coastal_clip(aoi, data, buffer=0)
    # Clip Coastal Buffer (by 100 Metres Intervals)

get_area_from_geojson(geojson_file)
    # AOI from GeoJson File (use geojson.io)

get_cloudless_mosaic(aoi, collection_name='sentinel-2-l2a', timeframe='2019-11-01/2022-11-31', cloudcover=10, resolution=100,
max=100, period='yearly', coastal_clip=False)
    # median composite - Cloudless Mosaic achieved combining images across time

get_country_admin_boundary(country, admin_type, admin)
    # AOI from Country Administrative Boundary

get_country_boundary(country)
    # AOI from a Country Nation Boundary

get_data(aoi, bands=[], collection_name='sentinel-2-l2a', timeframe='2023-01-01/2023-12-31', cloudcover=10, resolution=100,
max=<built-in function max>, period='monthly', coastal_clip=False)
    # Xarray Dataset from STAC by Period Yearly, Quarterly, Monthly, Weekly, Daily

get_evi(aoi, collection_name='sentinel-2-l2a', timeframe='2019-11-01/2022-11-31', cloudcover=10, resolution=100, max=100,
period='monthly', coastal_clip=False)
    # evi - Enhanced Vegetation

get_gci(aoi, collection_name='sentinel-2-l2a', timeframe='2019-11-01/2022-11-31', cloudcover=10, resolution=100, max=100,
period='monthly', coastal_clip=False)
    # gci - Green Chlorophyll Index

get_global_land_cover(aoi, name='io-lulc-9-class')
    # Get Global LandCover over AOI

get_landcover_mosaic(aoi, year, bands=['B02', 'B03', 'B04', 'B05', 'B06', 'B07', 'B08', 'B8A'], resolution=10, max=10000,
cloudcover=10, collection_name='sentinel-2-l2a', coastal_clip=False)
    # Generate Annual Landcover Mosaic with Multiple Bands for ML Classification

get_latest_images(aoi, collection_name='sentinel-2-l2a', timeframe='2023-01-01/2023-12-31', cloudcover=10, resolution=100,
max=100, period='daily', coastal_clip=False)
    # Latest RGB Images

get_ndmi(aoi, collection_name='sentinel-2-l2a', timeframe='2019-11-01/2022-11-31', cloudcover=10, resolution=100, max=100,
period='monthly', coastal_clip=False)
    # ndmi - Normalised Difference Moisture Index
```

```

get_ndvi(aoi, collection_name='sentinel-2-l2a', timeframe='2019-11-01/2022-11-31', cloudcover=10, resolution=100, max=100,
period='monthly', coastal_clip=False)
    # ndvi - Normalised Difference Vegetation Index

get_ndwi(aoi, collection_name='sentinel-2-l2a', timeframe='2019-11-01/2022-11-31', cloudcover=10, resolution=100, max=100,
period='monthly', coastal_clip=False)
    # ndmi - Normalised Difference Water Index

get_sipi(aoi, collection_name='sentinel-2-l2a', timeframe='2019-11-01/2022-11-31', cloudcover=10, resolution=100, max=100,
period='monthly', coastal_clip=False)
    # sipi - Structure Insensitive Pigment Index: which is helpful in early disease detection in vegetation.

init(type='local', maxWorkers=4, resolution=100)
    # Initialise and Configure Dask and Resolution Defaults

list_boundary_types(country)
    # List Administrative Boundaries In a Country

list_countries()
    # List Pacific Island Countries and Territories

list_country_boundary(country, admin_type)
    # List Areas/Locations of a Administration Type Within A Country

list_data_assets(collection_name)
    # List Data Assets (non-spectral) and Common Names within a Data Source, Pipeline or Sensor

list_data_bands(collection_name='sentinel-2-l2a')
    # List Data Bands and Common Names within a Data Source, Pipeline or Sensor

list_data_sources()
    # List Data Sources, Pipelines and Models

list_global_land_cover()
    # List Global LandCover DataSets

plot(data)
    # Plot Mean TimeSeries for Indices

plot_cloudiness(aoi, timeframe, collection_name='sentinel-2-l2a')
    # Plot Cloudiness Percentage Over AOI over Timeframe eg: "2020/2022"

save_multiple(data, file_name)
    # Save Multiple Outputs as GeoTIFF/COG Series

save_single(data, file_name)
    # Save Single Data as GeoTIFF/COG Series
    # eg: d = data.sel(time="2023-01-23").expand_dims(dim="time"); dep.save\_single(d, "tmp")

smooth(data)
    # Focal Mean Smoothing and Noise Removal

visualise(data, cmap=None)
    # Visual Data by Colour Maps

```

Data

```

__copyright__ = 'Pacific Community (SPC)'
__email__ = 'sachindras@spc.int'
__license__ = 'GPL'
__status__ = 'Development'
catalog = <Client id=microsoft-pc>
chunk_size = 4096
client = None
cluster = None
default_max = 100
default_resolution = 100
padm = country ... -176.24805 -13.28860))) [698 rows x 12 columns]

```

Author

Sachindra Singh