hda-pystac-client

April 22, 2024

1 DestinE Data Lake HDA PySTAC-Client connection from PolarTEP

1.1 Obtain DEDL Access Token to use the HDA service

[1]:
```
from getpass import getpass
import requests

HTTP_SUCCESS_CODE = 200

username = input()
password = getpass()
```

```
david.arthurs@polarview.org
........
```

[2]:
```
def get_auth_headers():
    # Fill in if you wish to get your access token automatically
    DEDL_IdP_token = 'https://identity.data.destination-earth.eu/auth/realms/
 ↪dedl/protocol/openid-connect/token'
    access_token_response = \
        requests.post(DEDL_IdP_token,
                      data = {'grant_type': 'password','scope' : 'openid',␣
 ↪'client_id' : 'hda-public', 'username' : username, 'password' : password},
                      headers = {"Content-Type" : "application/
 ↪x-www-form-urlencoded"})

    if(access_token_response.status_code == HTTP_SUCCESS_CODE):
        access_token = access_token_response.json()['access_token']

    return {'Authorization': 'Bearer {}'.format(access_token)}
```

1.2 Set username and password as environment variables to be used for DEDL data access

[3]:
```
import os

os.environ["EODAG__DEDL__AUTH__CREDENTIALS__USERNAME"] = username
os.environ["EODAG__DEDL__AUTH__CREDENTIALS__PASSWORD"] = password
```

2 Create pystac client object for HDA STAC API as in documentation

[4]:
```
from pystac_client import Client

HDA_API_URL = "https://hda.data.destination-earth.eu/stac"
cat = Client.open(HDA_API_URL, headers=get_auth_headers())
```

2.1 Query CMEMS available collections

this selection is based on collection list in documentation: https://destine-data-lake-docs.data.destination-earth.eu/en/latest/dedl-discovery-and-data-access/DestinE-Data-Portfolio/DestinE-Data-Portfolio.html?highlight=collections

[8]:
```
from rich.console import Console
import rich.table

console = Console()

hda_collections = cat.get_collections()

table = rich.table.Table(title="HDA collections", expand=True)
table.add_column("ID", style="cyan", justify="right",no_wrap=True)
table.add_column("Title", style="violet", no_wrap=True)
for collection in hda_collections:
    if "SEA_LEVEL" in collection.id:
        table.add_row(collection.id, collection.title)
console.print(table)
```

```
                                        HDA collections

                                     ID   Title                                                   ␣
↪

 EO.ECMWF.DAT.SEA_LEVEL_DAILY_GRIDDED_DATA…   Sea level gridded data from␣
↪satellite observations for the global…
 EO.ECMWF.DAT.SEA_LEVEL_DAILY_GRIDDED_DATA…   Sea level daily gridded data from␣
↪satellite observations for the …
```

```
EO.ECMWF.DAT.SEA_LEVEL_DAILY_GRIDDED_DATA…  Sea level daily gridded data from␣
↪satellite observations for the …
```

2.2 Obtain provider information for each individual collection

[17]:
```
table = rich.table.Table(title="HDA collections | Providers", expand=True)
table.add_column("Title", style="cyan", justify="right", no_wrap=True)
table.add_column("Provider", style="violet", no_wrap=True)

hda_collections = cat.get_collections()

for collection in hda_collections:
    if "SEA_LEVEL" in collection.id:
        collection_details = cat.get_collection(collection.id)
        provider = ','.join(str(x.name) for x in collection_details.providers)
        table.add_row(collection_details.title, provider)
console.print(table)
```

```
                                        HDA collections | Providers

                                                 Title   Provider          ␣
↪

 Sea level gridded data from satellite observations for t…  European Centre for␣
↪Medium-Range Weather Forecasts…
 Sea level daily gridded data from satellite observations…  European Centre for␣
↪Medium-Range Weather Forecasts…
 Sea level daily gridded data from satellite observations…  European Centre for␣
↪Medium-Range Weather Forecasts…
```

2.3 Inspect Items of a Collection

The main functions for getting items return iterators, where pystac-client will handle retrieval of additional pages when needed. Note that one request is made for the first ten items, then a second request for the next ten.

[19]:
```
coll_name = 'EO.ECMWF.DAT.
 ↪SEA_LEVEL_DAILY_GRIDDED_DATA_FOR_BLACK_SEA_1993_PRESENT'
search = cat.search(
    max_items=10,
    collections=[coll_name],
    bbox=[27,40.5,41,46],
    datetime="2023-09-09T00:00:00Z/2023-09-20T23:59:59Z"
)
```

```
coll_items = search.item_collection()
console.print(f"For collection {coll_name} we found {len(coll_items)} items")
```

```
For collection EO.ECMWF.DAT.
↪SEA_LEVEL_DAILY_GRIDDED_DATA_FOR_BLACK_SEA_1993_PRESENT we found 1 items
```

[20]:
```
import geopandas

df = geopandas.GeoDataFrame.from_features(coll_items.to_dict(), crs="epsg:4326")
df.head()
```

[20]:
```
                                              geometry  \
0  POLYGON ((27.00000 40.50000, 27.00000 46.00000…

                                           providers              datetime  \
0  [{'name': 'copernicus_climate_data_store', 'de…  2023-09-09T00:00:00Z

  start_datetime end_datetime instruments               sar:product_type
0     2023-09-09   2023-09-20      [None]  SATELLITE_SEA_LEVEL_BLACK_SEA
```

2.4 Inspect STAC assets of an item

[12]:
```
import rich.table

selected_item = coll_items[0]

table = rich.table.Table(title="Assets in STAC Item")
table.add_column("Asset Key", style="cyan", no_wrap=True)
table.add_column("Description")
for asset_key, asset in selected_item.assets.items():
    table.add_row(asset_key, asset.title)

console.print(table)
```

```
     Assets in STAC Item

 Asset Key      Description

 downloadLink   Download link
```

[13]:
```
down_uri = selected_item.assets["downloadLink"].href
console.print(f"Download link of asset is {down_uri}")
```

Download link of asset is
https://hda.data.destination-earth.eu/stac/collections/EO.ECMWF.DAT.SEA_LEVEL_DAILY_GRIDDED_DAT
_PRESENT/items/SATELLITE_SEA_LEVEL_BLACK_SEA_20230909_20230920_eb2c671be7501202950d00d9886db6e3
ovider=copernicus_climate_data_store&_dc_qs=%257B%2522area%2522%253A%252B%252241.0%2F-72.5%2F40
%252B%2522day%2522%253A%252B%255B%252215%2522%252C%252B%252217%2522%252C%252B%252220%2522%252C
C%252B%252213%2522%252C%252B%252214%2522%252C%252B%252216%2522%252C%252B%252219%2522%252C%252B
B%252211%2522%252C%252B%252212%2522%252C%252B%252218%2522%255D%252C%252B%2522format%2522%253A%2
C%252B%2522month%2522%253A%252B%252209%2522%252C%252B%2522year%2522%253A%252B%25222023%2522%257

2.4.1 Download asset to JupyterLab

[14]:
```
selected_item.id
```

[14]:
```
'SATELLITE_SEA_LEVEL_BLACK_SEA_20230909_20230920_eb2c671be7501202950d00d9886db6e
3123aea4a'
```

[15]:
```
selected_item.assets["downloadLink"]
```

[15]:
```
<Asset href=https://hda.data.destination-earth.eu/stac/collections/EO.ECMWF.DAT.
SEA_LEVEL_DAILY_GRIDDED_DATA_FOR_BLACK_SEA_1993_PRESENT/items/SATELLITE_SEA_LEVE
L_BLACK_SEA_20230909_20230920_eb2c671be7501202950d00d9886db6e3123aea4a/download?
provider=copernicus_climate_data_store&_dc_qs=%257B%2522area%2522%253A%252B%2522
41.0%2F-72.5%2F40.5%2F-72.0%2522%252C%252B%2522day%2522%253A%252B%255B%252215%25
22%252C%252B%252217%2522%252C%252B%252220%2522%252C%252B%252210%2522%252C%252B%2
52213%2522%252C%252B%252214%2522%252C%252B%252216%2522%252C%252B%252219%2522%252
C%252B%252209%2522%252C%252B%252211%2522%252C%252B%252212%2522%252C%252B%252218%
2522%255D%252C%252B%2522format%2522%253A%252B%2522zip%2522%252C%252B%2522month%2
522%253A%252B%252209%2522%252C%252B%2522year%2522%253A%252B%25222023%2522%257D>
```

[16]:
```
# Make http request for remote file data
data = requests.get(selected_item.assets["downloadLink"].href,
                    headers=get_auth_headers())
mtype = selected_item.assets["downloadLink"].media_type.split("/")[1]
# Save file data to local copy
with open(f"{selected_item.id}.{mtype}", 'wb')as file:
    file.write(data.content)
```

[ ]:
```

```