

How To Access Data With Python

The following Python code example demonstrates how to configure a connection to download data from an Earthdata Login enabled server. Note that you will need to a secure way to configure the Earthdata Login username and password.

```
#!/usr/bin/python
from cookielib import CookieJar
from urllib import urlencode

import urllib2

# The user credentials that will be used to authenticate access to the data

username = "<Your Earthdata login username>"
password = "<Your Earthdata login password>"

# The url of the file we wish to retrieve

url = "http://e4ftl01.cr.usgs.gov/MOLA/MYD17A3H.006/2009.01.01/MYD17A3H.A2009001.h12v05.006.2015198130546.hdf.xml"

# Create a password manager to deal with the 401 reponse that is returned from
# Earthdata Login

password_manager = urllib2.HTTPPasswordMgrWithDefaultRealm()
password_manager.add_password(None, "https://urs.earthdata.nasa.gov", username, password)

# Create a cookie jar for storing cookies. This is used to store and return
# the session cookie given to use by the data server (otherwise it will just
# keep sending us back to Earthdata Login to authenticate). Ideally, we
# should use a file based cookie jar to preserve cookies between runs. This
# will make it much more efficient.

cookie_jar = CookieJar()

# Install all the handlers.

opener = urllib2.build_opener(
    urllib2.HTTPBasicAuthHandler(password_manager),
    #urllib2.HTTPHandler(debuglevel=1),    # Uncomment these two lines to see
    #urllib2.HTTPSHandler(debuglevel=1),  # details of the requests/responses
    urllib2.HTTPCookieProcessor(cookie_jar))
urllib2.install_opener(opener)

# Create and submit the request. There are a wide range of exceptions that
# can be thrown here, including HTTPError and URLError. These should be
# caught and handled.

request = urllib2.Request(url)
response = urllib2.urlopen(request)

# Print out the result (not a good idea with binary data!)

body = response.read()
print body
```

Here is another example that will only work with later versions of python:

```
#!/usr/bin/python
```

```

import requests # get the requests library from https://github.com/requests/requests

# overriding requests.Session.rebuild_auth to maintain headers when redirected

class SessionWithHeaderRedirection(requests.Session):

    AUTH_HOST = 'urs.earthdata.nasa.gov'

    def __init__(self, username, password):

        super().__init__()

        self.auth = (username, password)

# Overrides from the library to keep headers when redirected to or from
# the NASA auth host.

def rebuild_auth(self, prepared_request, response):

    headers = prepared_request.headers

    url = prepared_request.url

    if 'Authorization' in headers:

        original_parsed = requests.utils.urlparse(response.request.url)

        redirect_parsed = requests.utils.urlparse(url)

        if (original_parsed.hostname != redirect_parsed.hostname) and \

            redirect_parsed.hostname != self.AUTH_HOST and \

            original_parsed.hostname != self.AUTH_HOST:

            del headers['Authorization']

    return

# create session with the user credentials that will be used to authenticate access to the data

username = "USERNAME"

password= "PASSWORD"

session = SessionWithHeaderRedirection(username, password)

# the url of the file we wish to retrieve

url = "http://e4ftl01.cr.usgs.gov/MOLA/MYD17A3H.006/2009.01.01/MYD17A3H.A2009001.h12v05.006.2015198130546.hdf.xml"

```

```

# extract the filename from the url to be used when saving the file
filename = url[url.rfind('/')+1:]

try:

    # submit the request using the session

    response = session.get(url, stream=True)

    print(response.status_code)


    # raise an exception in case of http errors

    response.raise_for_status()


    # save the file

    with open(filename, 'wb') as fd:

        for chunk in response.iter_content(chunk_size=1024*1024):

            fd.write(chunk)

except requests.exceptions.HTTPError as e:

    # handle any errors here

    print(e)

```

Here is another simple example:

```

# assuming variables `username`, `password` and `url` are set...

# Example URL

url = "https://n5eil01u.ecs.nsidc.org/MOST/MOD10A1.006/2016.12.31/"

# url = "https://e4ftl01.cr.usgs.gov/MOTA/MCD43A2.006/2017.09.04/"

import requests

with requests.Session() as session:

    s.auth = (username, password)

    r1 = session.request('get', url)

    r = session.get(r1.url, auth=(username, password))

    if r.ok:

        print r.content # Say

```

Accessing Data from NSIDC:

NSIDC has provided sample scripts to access their data with Python:



`NSIDC_SingleDL.py`



`NSIDC_Parse_HTML_BatchDL.py`