
pyNEID

Release 0.1.0

Mihseh Kong

May 04, 2021

CONTENTS:

1	API	3
1.1	Archive Class	3
1.2	NeidTap Class	8
1.3	TapJob Class	8
1.4	Object Lookup Class	8
2	Indices and tables	9
	Index	11

This package is designed to be an API interface to the [NEID Data Archive](#).

With pyNEID you can:

- log in using your NEID Archive credentials, or use without login for access to public data only
- query the archive for tables of L0, L1, and L2 metadata
- download FITS files of any level
- more features to come...

The package may be installed via pip:

```
$ pip install git+https://github.com/Caltech-IPAC/pyNEID.git
```

Or by downloading directly from the [GitHub repository](#).

The *Archive class* contains many useful methods for working with the NIED archive.

This package (alpha-release) is under active development by the NASA Exoplanet Science Institute.

Please report any bugs, issues, or feature requests to the [NEID Help Desk](#).

1.1 Archive Class

class pyneid.neid.**Archive** (***kwargs*)

‘Archive’ class provides NEID archive access functions for searching NEID data via TAP interface.

The user’s NEID credentials (given at login as a cookie file or a token string) are used to search the proprietary data.

Parameters **debugfile** – a file path for the debug output

In the following examples, “neid” will represent an instance of the Archive class:

Examples

```
>>> import os
>>> import sys
```

```
>>> from pyneid.neid import Neid
```

```
>>> neid = Archive()
```

download (*metapath, datalevel, format, outdir, **kwargs*)

The download method allows users to download FITS files shown in the retrieved metadata file. The column ‘filepath’ must be included in the metadata file columns in order to download files.

Parameters

- **metapath** (*string*) – a full path metadata table obtained from running query methods
- **datalevel** (*string*) – 10, 11, 12, eng, solar10, solar11, solar12, solareng
- **datetime** (*string*) – a datetime range string, see *query_datetime*
- **format** (*string*) – metasata table’s format: ipac, votable, csv, or tsv.
- **outdir** (*string*) – the directory for depositing the returned files
- **cookiepath** (*string*) – (optional) cookie file path for downloading the proprietary NEID data.
- **token** (*string*) – (optional) token string obtained from login.
- **start_row** (*integer*) – (optional) starting row

- **end_row** (*integer*) – (optional) ending row
- **calibfile** (*bool*) – whether to download the associated calibration files (0/1); default is False.

login (***kwargs*)

login method validates a user has a valid NEID account; it takes two ‘keyword’ arguments: userid and password. If the inputs are not provided in the keyword, the auth method prompts for inputs.

login method returns both cookie header and a token string in the returned message file.

If cookiepath is provided, the cookie will be saved to the cookiepath.

The token string will be saved in the variable “token” in memory to be used for other Archive methods in the current session.

Parameters **userid** – a valid user id in the NEID’s user table.

Examples

```
# without prompt >>> neid.login(userid='xxxx', password='xxxxxx', cookiepath=cookiepath) # or >>>
neid.login(userid='xxxx', password='xxxxxx', token=token_string) # or # prompt for userid and password
>>> neid.login(cookiepath=cookiepath) >>> neid.login()
```

query_adql (*query, **kwargs*)

‘query_adql’ method receives a qualified ADQL query string from user input.

Parameters

- **query** (*string*) – a ADQL query
- **outpath** (*string*) – (optional) a full output filepath of the returned metadata table
- **cookiepath** (*string*) – (optional) a full cookie file path saved from login for querying the proprietary NEID data.
- **token** (*string*) – (optional) a token string save in memory from login for querying the proprietary NEID data; the token is only valid for the current session.
- **format** (*string*) – (optional) votable, ipac, csv, tsv (default: ipac) maxrec (integer): (optional) maximum records to be returned

default: -1 or not specified will return all requested records

query_criteria (*param, **kwargs*)

‘query_criteria’ method allows the search of NEID data by multiple the parameters specified in a dictionary (param).

Parameters

- **param** (*dict*) – a dictionary containing the following keys; datalevel, datetime, position, target. See individual methods for input forming.
- **outpath** (*string*) – (optional) a full output filepath of the returned metadata table
- **cookiepath** (*string*) – (optional) a full cookie file path saved from login for querying the proprietary NEID data.
- **token** (*string*) – (optional) a token string save in memory from login for querying the proprietary NEID data; the token is only valid for the current session.

- **format** (*string*) – (optional) votable, ipac, csv, tsv (default: ipac) maxrec (integer): (optional) maximum records to be returned
default: -1 or not specified will return all requested records

query_datetime (*datalevel, datetime, **kwargs*)
‘query_datetime’ method search NEID data by ‘datetime’ range

Parameters

- **datalevel** (*string*) – 10, 11, 12, eng, solarl0, solarl1, solarl2, solareng
- **datetime** (*string*) – a datetime string in the format of datetime1/datetime2 where ‘/’ separates the two datetime values` of format ‘yyyy-mm-dd hh:mm:ss’
- **outpath** (*string*) – (optional) a full output filepath of the returned metadata table
- **cookiepath** (*string*) – (optional) a full cookie file path saved from login for querying the proprietary NEID data.
- **token** (*string*) – (optional) a token string save in memory from login for querying the proprietary NEID data; the token is only valid for the current session.
- **format** (*string*) – (optional) Output format: votable, ipac, csv, tsv (default: votable)
- **maxrec** (*integer*) – (optional) maximum records to be returned default: -1 or not specified will return all requested records

Note: The following inputs are acceptable for the datetime parameter:

‘datetime1/’: will search data with datetime later than (\geq) datetime1

‘/datetime2’: will search data with datetime earlier than (\leq) datetime2

‘datetime1’: will search data with datetime equal to ($=$) datetime1 (this is not recommended)

Examples

```
>>> datalevel = '10',
>>> datetime = '2020-11-16 06:10:55/2020-11-18 00:00:00'
```

```
>>> datalevel = '11',
>>> datetime = '2020-11-16 06:10:55/'
```

```
>>> datalevel = '12',
>>> datetime = '/2020-11-18 00:00:00'
```

query_object (*datalevel, object, **kwargs*)
‘query_object’ method search NEID data by ‘object name’. This method resolves the object name into coordinates to be used as the center of the circle position search with default radius of 0.5 deg.

Parameters

- **datalevel** – 10, 11, 12, eng, solarl0, solarl1, solarl2, solareng
- **object** (*string*) – an object name resolvable by SIMBAD, NED, and ExoPlanet’s name_resolve; outpath (*string*): (optional) a full output filepath of the returned metadata table

- **cookiepath** (*string*) – (optional) a full cookie file path saved from login for querying the proprietary NEID data.
- **token** (*string*) – (optional) a token string save in memory from login for querying the proprietary NEID data; the token is only valid for the current session.
- **format** (*string*) – (optional) votable, ipac, csv, tsv (default: ipac) maxrec (integer): (optional) maximum records to be returned
default: -1 or not specified will return all requested records

Examples

datalevel = '11', object = 'WD 1145+017'

query_piname (*datalevel*, *piname*, ***kwargs*)
'query_piname' method search NEID data by PI name

Parameters

- **datalevel** (*string*) – l0, l1, l2, eng, solarl0, solarl1, solarl2, solareng
- **datetime** (*string*) – a datetime string in the format of
- **piname** (*string*) – PI name as formatted in the project's catalog
- **outpath** (*string*) – (optional) a full output filepath of the returned metadata table
- **cookiepath** (*string*) – (optional) a full cookie file path saved from login for querying the proprietary NEID data.
- **token** (*string*) – (optional) a token string save in memory from login for querying the proprietary NEID data; the token is only valid for the current session.
- **format** (*string*) – (optional) votable, ipac, csv, tsv (default: ipac) maxrec (integer): (optional) maximum records to be returned
default: -1 or not specified will return all requested records

query_position (*datalevel*, *position*, ***kwargs*)
'query_position' method search NEID data by 'position'

Parameters

- **datalevel** (*string*) – l0, l1, l2, eng, solarl0, solarl1, solarl2, solareng
- **datetime** (*string*) – a datetime string as specified in *query_datetime*
- **position** (*string*) – a position string in the format of outpath (string): (optional) a full output filepath of the returned metadata table
- **cookiepath** (*string*) – (optional) a full cookie file path saved from login for querying the proprietary NEID data.
- **token** (*string*) – (optional) a token string save in memory from login for querying the proprietary NEID data; the token is only valid for the current session.
- **format** (*string*) – (optional) votable, ipac, csv, tsv (default: ipac) maxrec (integer): (optional) maximum records to be returned
default: -1 or not specified will return all requested records

Note:

Position can be a string in the following formats

1. circle ra dec radius;
2. polygon ra1 dec1 ra2 dec2 ra3 dec3 ra4 dec4;
3. box ra dec width height;

All ra dec should be specified in J2000 coordinates.

Examples

```
>>> datalevel = 'l1',
>>> position = 'circle 230.0 45.0 0.5'
```

query_program (*datalevel*, *program*, ***kwargs*)
 ‘query_program’ method search NEID data by ‘program’

Parameters

- **datalevel** (*string*) – l0, l1, l2, eng, solarl0, solarl1, solarl2, solareng
- **datetime** (*string*) – datetime string
- **program** (*string*) – program ID in the project’s catalog
- **outpath** (*string*) – (optional) a full output filepath of the returned metadata table
- **cookiepath** (*string*) – (optional) a full cookie file path saved from login for querying the proprietary NEID data.
- **token** (*string*) – (optional) a token string save in memory from login for querying the proprietary NEID data; the token is only valid for the current session.
- **format** (*string*) – (optional) votable, ipac, csv, tsv (default: ipac) maxrec (integer): (optional) maximum records to be returned

default: -1 or not specified will return all requested records

query_qobject (*datalevel*, *qobject*, ***kwargs*)
 ‘query_qobject’ method search NEID data for ‘qobject’ column value. This method resolves the object name into coordinates to be used as the

center of the circle position search with default radius of 0.5 deg.

Parameters

- **datalevel** – l0, l1, l2, eng, solarl0, solarl1, solarl2, solareng
- **qobject** (*string*) – an object name as specified in the QOBJECT column. This is usually the Gaia DR2 ID outpath (string): (optional) a full output filepath of the returned metadata table
- **cookiepath** (*string*) – (optional) a full cookie file path saved from login for querying the proprietary NEID data.
- **token** (*string*) – (optional) a token string save in memory from login for querying the proprietary NEID data; the token is only valid for the current session.
- **format** (*string*) – (optional) votable, ipac, csv, tsv (default: ipac) maxrec (integer): (optional) maximum records to be returned

default: -1 or not specified will return all requested records

1.2 NeidTap Class

class pyneid.neid.NeidTap(*url*, ***kwargs*)

NeidTap class provides client access to NEID's TAP service. Public data doesn't not require user login, optional NEID login via NeidLogin class are used to search a user's proprietary data.

Parameters

- **query** (*string*) – a SQL statement in specified query language request (string): (optional) default 'doQuery' lang (string): (optional) default 'ADQL' phase (string): (optional) default 'RUN' format (string): (optional) default 'votable' maxrec (int): (optional) default '2000'
- **cookiefile** (*string*) – a full path cookie file containing user info
- **debug** (*bool*) – default False

Examples

```
>>> service = NeidTap(url, cookiefile=cookiepath)
# or
>>> service = NeidTap(url)
# or
>>> job = service.send_async (query, format='votable', request='doQuery', ...)
# or
>>> job = service.send_sync (query, format='votable', request='doQuery', ...)
```

get_data (*resultpath*)

loop until job is complete, then download the data to the given resultpath

msg

tapjob contains async job's status; resulttbl is the result of sync saved an astropy table

1.3 TapJob Class

class pyneid.neid.TapJob(*statusurl*, ***kwargs*)

TapJob class is used internally by TapClient class to store a Tap job's parameters and returned job status and result urls.

1.4 Object Lookup Class

class pyneid.neid.objLookup(*object*, ***kwargs*)

objLookup wraps ExoPlanet's web name resolver into a python class; the exoLookup checks the exoplanet archive database and if that fails it checks with the Sesame web service at CDS. Sesame checks the CDS database and if that fails it checks NED. So this class covers SIMBAD, NED, and ExoPlanet search.

Parameters **object** (*char*) – object name to be resolved

debug = 0

{ objLookup.init

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

INDEX

A

Archive (*class in pyneid.neid*), 3

D

debug (*pyneid.neid.objLookup attribute*), 8

download() (*pyneid.neid.Archive method*), 3

G

get_data() (*pyneid.neid.NeidTap method*), 8

L

login() (*pyneid.neid.Archive method*), 4

M

msg (*pyneid.neid.NeidTap attribute*), 8

N

NeidTap (*class in pyneid.neid*), 8

O

objLookup (*class in pyneid.neid*), 8

Q

query_adql() (*pyneid.neid.Archive method*), 4

query_criteria() (*pyneid.neid.Archive method*), 4

query_datetime() (*pyneid.neid.Archive method*), 5

query_object() (*pyneid.neid.Archive method*), 5

query_piname() (*pyneid.neid.Archive method*), 6

query_position() (*pyneid.neid.Archive method*), 6

query_program() (*pyneid.neid.Archive method*), 7

query_qobject() (*pyneid.neid.Archive method*), 7

T

TapJob (*class in pyneid.neid*), 8