pyobis Documentation

Release 0.1.0

Scott Chamberlain

December 13, 2016

Contents

1 Installation **3**

2 library API **5**

3 Taxa module **7**

4 Occurrence module **9**

5 Resources module **11**

6 Groups module **13**

7 Ndes module **15**

8 Checklist module **17**

9 Meta **19**

9.1 Contents 19

9.2 License 28

9.3 Indices and tables 28

Python Module Index **29**

Python client for the OBIS API.

Source on GitHub at sckott/pyobis

Other OBIS clients:

- R: *robis*, iobis/robis

CHAPTER 1

Installation

from pypi

```
pip install pyobis
```

dev version

```
pip install git+git://github.com/sckott/pyobis.git#egg=pyobis
```

CHAPTER 2

library API

pyobis is split up into modules for each of the groups of API methods.

- *taxa* - Taxonomic names
- *occurrences* - Occurrence search, and occurrence downloads
- *resources* - Resources
- *groups* - Groups
- *nodes* - Nodes
- *checklist* - Checklist

You can import the entire library, or each module individually as needed.

CHAPTER 3

Taxa module

```
from pyobis import taxa
taxa.search(scientificname = 'Mola mola')
taxa.search(scientificname = 'Mola mola', offset=10, limit=10)
taxa.search(geometry='POLYGON((30.1 10.1, 10 20, 20 40, 40 40, 30.1 10.1))', limit=20)
taxa.search(aphiaid=key, year="2013", limit=20)
taxa.taxon(406296)
taxa.taxon(415282)
```

CHAPTER 4

Occurrence module

Search

```
from pyobis import occurrences
occurrences.search(scientificname = 'Mola mola')
occurrences.search(scientificname = 'Mola mola', offset=0, limit=10)
occurrences.search(geometry='POLYGON((30.1 10.1, 10 20, 20 40, 40 40, 30.1 10.1))', limit=20)
occurrences.search(aphiaid=key, year="2013", limit=20)
```

Download

```
res = occ.download(year = 2001, scientificname = 'Orcinus')
res.uuid
res.status()
res.fetch()
```

CHAPTER 5

Resources module

```
from pyobis import resources
resources.search(scientificname = ['Mola', 'Abra', 'Lanice', 'Pectinaria'])
resources.resource(103)
resources.citation(scientificname = 'Mola mola')
```

CHAPTER 6

Groups module

```
from pyobis import groups
groups.group()
groups.group(limit = 3)
```

CHAPTER 7

Ndes module

```
from pyobis import nodes
nodes.node()
```

CHAPTER 8

Checklist module

```
from pyobis import checklist as ch
ch.list(year = 2005, scientificname = 'Cetacea')
```

CHAPTER 9

Meta

- License: MIT, see LICENSE file
- Please note that this project is released with a Contributor Code of Conduct. By participating in this project you agree to abide by its terms.

9.1 Contents

9.1.1 occurrences module

`occurrences.search`(*scientificname=None*, *aphiaid=None*, *obisid=None*, *resourceid=None*, *startdate=None*, *enddate=None*, *startdepth=None*, *enddepth=None*, *geometry=None*, *year=None*, *qc=None*, *fields=None*, *limit=500*, *offset=0*, ***kwargs*)

Search OBIS occurrences

Parameters

- **aphiaid** – [Fixnum] A obis occurrence identifier
- **scientificname** – [String,Array] One or more scientific names from the OBIS backbone. All included and synonym taxa are included in the search.
- **year** – [Fixnum] The 4 digit year. A year of 98 will be interpreted as AD 98. Supports range queries, smaller,larger (e.g., '1990,1991', whereas '1991,1990' wouldn't work)
- **geometry** – [String] Well Known Text (WKT). A WKT shape written as either POINT, LINESTRING, LINEARRING or POLYGON. Example of a polygon: ((30.1 10.1, 20, 20 40, 40 40, 30.1 10.1)) would be queried as http://bit.ly/1BzNwDq
- **obisid** – [Fixnum] An OBIS id. This is listed as the *id* or *valid_id* in *taxa/taxon* results
- **aphiaid** – [Fixnum] An Aphia id. This is listed as the *worms_id* in *taxa/taxon* results
- **resourceid** – [Fixnum] An resource id
- **startdate** – [Fixnum] Start date
- **enddate** – [Boolean] End date
- **startdepth** – [Fixnum] Start depth
- **enddepth** – [Boolean] End depth
- **qc** – [String] Quality control flags
- **fields** – [Array] Array of field names

- **limit** – [Fixnum] Number of results to return. Default: 1000
- **offset** – [Fixnum] Start at record. Default: 0

Returns A dictionary

Usage:

```
from pyobis import occurrences as occ
occ.search(scientificname = 'Mola mola')

# Many names
occ.search(scientificname = ['Mola', 'Abra', 'Lanice', 'Pectinaria'])

# Use paging parameters (limit and start) to page. Note the different results
# for the two queries below.
occ.search(scientificname = 'Mola mola', offset=0, limit=10)
occ.search(scientificname = 'Mola mola', offset=10, limit=10)

# Search on a bounding box
## in well known text format
occ.search(geometry='POLYGON((30.1 10.1, 10 20, 20 40, 40 40, 30.1 10.1))', limit=20)
from pyobis import taxa
res = taxa.search(scientificname='Mola mola')['results'][0]
occ.search(obisid=res['id'], geometry='POLYGON((30.1 10.1, 10 20, 20 40, 40 40, 30.1 10.1))', li
occ.search(aphiaid=res['worms_id'], geometry='POLYGON((30.1 10.1, 10 20, 20 40, 40 40, 30.1 10.1

# Get occurrences for a particular eventDate
occ.search(aphiaid=res['worms_id'], year="2013", limit=20)
```

`occurrences.get`(*id*, ***kwargs*)

Get an OBIS occurrence

Parameters **id** – [Fixnum] An obis occurrence identifier

Returns A dictionary

Usage:

```
from pyobis import occurrences as occ
occ.get(id = 14333)
occ.get(id = 135355)

# many at once
[ occ.get(id = x) for x in [14333, 135355, 276413] ]
```

`occurrences.download`(*scientificname=None*, *aphiaid=None*, *obisid=None*, *resourceid=None*, *startdate=None*, *enddate=None*, *startdepth=None*, *enddepth=None*, *geometry=None*, *year=None*, *qc=None*, *fields=None*, ***kwargs*)

Download OBIS occurrences

Parameters

- **aphiaid** – [Fixnum] A obis occurrence identifier
- **scientificname** – [String,Array] One or more scientific names from the OBIS backbone. All included and synonym taxa are included in the search.
- **year** – [Fixnum] The 4 digit year. A year of 98 will be interpreted as AD 98. Supports range queries, smaller,larger (e.g., '1990,1991', whereas '1991,1990' wouldn't work)
- **geometry** – [String] Well Known Text (WKT). A WKT shape written as either POINT, LINESTRING, LINEARRING or POLYGON. Example of a polygon: ((30.1 10.1, 20, 20

40, 40 40, 30.1 10.1)) would be queried as http://bit.ly/1BzNwDq

- **obisid** – [Fixnum] An OBIS id. This is listed as the *id* or *valid_id* in *taxa/taxon* results
- **aphiaid** – [Fixnum] An Aphia id. This is listed as the *worms_id* in *taxa/taxon* results
- **resourceid** – [Fixnum] An resource id
- **startdate** – [Fixnum] Start date
- **enddate** – [Boolean] End date
- **startdepth** – [Fixnum] Start depth
- **enddepth** – [Boolean] End depth
- **qc** – [String] Quality control flags
- **fields** – [Array] Array of field names

Returns An object of class ObisDownload with methods to continue accessing the data

Usage:

```
from pyobis import occurrences as occ

# query to generate a download job
res = occ.download(year = 2001, scientificname = 'Orcinus')

# get the uuid for your download job
res.uuid

# get status of download prep
res.status()

# fetch file, writes to disk
res.fetch()

# get file path
x.file_path

# unzip the file
import zipfile
import tempfile
import shutil
import os

zipf = zipfile.ZipFile(x, 'r')
dir = tempfile.mkdtemp()
zipf.extractall(dir)
fpath = dir + '/' + os.listdir(dir)[0]
zipf.close()

# read some lines of the csv
import csv
with open(fpath) as csvfile:
    reader = csv.DictReader(csvfile)
    for row in reader:
        print(row['genus'], row['species'])

# cleanup
shutil.rmtree(dir)
```

`occurrences.ObisDownload()`

ObisDownload class

methods:

- uuid: get uuid for the download
- status: get download status
- fetch: retrieve the download

9.1.2 taxa module

`taxa.search(scientificname=None, aphiaid=None, obisid=None, resourceid=None, startdate=None, enddate=None, startdepth=None, enddepth=None, geometry=None, year=None, fields=None, limit=500, offset=0, **kwargs)`

Search OBIS taxa

Parameters

- **aphiaid** – [Fixnum] A obis occurrence identifier
- **scientificname** – [String,Array] One or more scientific names from the OBIS backbone. All included and synonym taxa are included in the search.
- **year** – [Fixnum] The 4 digit year. A year of 98 will be interpreted as AD 98. Supports range queries, smaller,larger (e.g., '1990,1991', whereas '1991,1990' wouldn't work)
- **geometry** – [String] Well Known Text (WKT). A WKT shape written as either POINT, LINESTRING, LINEARRING or POLYGON. Example of a polygon: ((30.1 10.1, 20, 20 40, 40 40, 30.1 10.1)) would be queried as http://bit.ly/1BzNwDq}.
- **obisid** – [Fixnum] An OBIS id. This is listed as the *id* or *valid_id* in *taxa/taxon* results
- **aphiaid** – [Fixnum] An Aphia id. This is listed as the *worms_id* in *taxa/taxon* results
- **resourceid** – [Fixnum] An resource id
- **startdate** – [Fixnum] Start date
- **enddate** – [Boolean] End date
- **startdepth** – [Fixnum] Start depth
- **enddepth** – [Booean] End depth
- **fields** – [Array] Array of field names
- **limit** – [Fixnum] Number of results to return. Default: 1000
- **offset** – [Fixnum] Start at record. Default: 0

Returns A dictionary

Usage:

```
from pyobis import taxa
taxa.search(scientificname = 'Mola mola')

# Use paging parameters (limit and start) to page. Note the different results
# for the two queries below.
taxa.search(scientificname = 'Mola mola', offset=0, limit=10)
taxa.search(scientificname = 'Mola mola', offset=10, limit=10)

# Search on a bounding box
```

```
## in well known text format
taxa.search(geometry='POLYGON((30.1 10.1, 10 20, 20 40, 40 40, 30.1 10.1))', limit=20)
from pyobis import taxa
key = taxa.search(query='Mola mola')[0]['key']
taxa.search(aphiaid=key, geometry='POLYGON((30.1 10.1, 10 20, 20 40, 40 40, 30.1 10.1))', limit=

# Get taxon for a particular eventDate
taxa.search(aphiaid=key, year="2013", limit=20)
```

`taxa.taxon`(*id*, ***kwargs*)

Get taxon by ID

Parameters **id** – [Fixnum] An OBIS taxon identifier

Returns A dictionary

Usage:

```
from pyobis import taxa
taxa.taxon(545439)
taxa.taxon(402913)
taxa.taxon(406296)
taxa.taxon(415282)
```

`taxa.common`(*id*, ***kwargs*)

Get common names for a taxon by ID

Parameters **id** – [Fixnum] An OBIS taxon identifier. Required

Returns A dictionary

Usage:

```
from pyobis import taxa
# have common names
taxa.common(402913)
taxa.common(406296)
# no common names
taxa.common(415282)
```

`taxa.taxon_search`(*scientificname=None*, *aphiaid=None*, *obisid=None*, ***kwargs*)

Get taxon by ID

Parameters **id** – [Fixnum] An OBIS taxon identifier

Returns A dictionary

Usage:

```
from pyobis import taxa
taxa.taxon_search(scientificname = 'Mola mola')
taxa.taxon_search(scientificname = 'Mola')
taxa.taxon_search(aphiaid = 127405)
taxa.taxon_search(obisid = 472375)
```

9.1.3 groups module

`groups.group`(*limit=100*, *offset=0*, ***kwargs*)

Get OBIS groups

Groups are taxonomic groups

Parameters

- **limit** – [Fixnum] Number of results to return. Default: 1000
- **offset** – [Fixnum] Start at record. Default: 0

Returns A dictionary

Usage:

```
from pyobis import groups
groups.group()
groups.group(limit = 3)
groups.group(limit = 3, offset = 1)
```

9.1.4 resources module

`resources.search`(*scientificname=None*, *aphiaid=None*, *obisid=None*, *startdate=None*, *enddate=None*, *startdepth=None*, *enddepth=None*, *geometry=None*, *year=None*, *limit=500*, *offset=0*, ***kwargs*)

Search OBIS resources

Parameters

- **aphiaid** – [Fixnum] A obis occurrence identifier
- **scientificname** – [String,Array] One or more scientific names from the OBIS backbone. All included and synonym taxa are included in the search.
- **year** – [Fixnum] The 4 digit year. A year of 98 will be interpreted as AD 98. Supports range queries, smaller,larger (e.g., '1990,1991', whereas '1991,1990' wouldn't work)
- **geometry** – [String] Well Known Text (WKT). A WKT shape written as either POINT, LINESTRING, LINEARRING or POLYGON. Example of a polygon: ((30.1 10.1, 20, 20 40, 40 40, 30.1 10.1)) would be queried as http://bit.ly/1BzNwDq
- **obisid** – [Fixnum] An OBIS id. This is listed as the *id* or *valid_id* in *taxa/taxon* results
- **aphiaid** – [Fixnum] An Aphia id. This is listed as the *worms_id* in *taxa/taxon* results
- **startdate** – [Fixnum] Start date
- **enddate** – [Boolean] End date
- **startdepth** – [Fixnum] Start depth
- **enddepth** – [Boolean] End depth
- **limit** – [Fixnum] Number of results to return. Default: 1000
- **offset** – [Fixnum] Start at record. Default: 0

Returns A dictionary

Usage:

```
from pyobis import resources
resources.search(scientificname = 'Mola mola')

# Many names
resources.search(scientificname = ['Mola', 'Abra', 'Lanice', 'Pectinaria'])

# Use paging parameters (limit and start) to page. Note the different results
# for the two queries below.
```

```
resources.search(scientificname = 'Mola mola', offset=0, limit=3)
resources.search(scientificname = 'Mola mola', offset=10, limit=2)

# Search on a bounding box
## in well known text format
resources.search(geometry='POLYGON((30.1 10.1, 10 20, 20 40, 40 40, 30.1 10.1))', limit=20)
from pyobis import taxa
res = taxa.search(scientificname='Mola mola')['results'][0]
resources.search(obisid=res['id'], geometry='POLYGON((30.1 10.1, 10 20, 20 40, 40 40, 30.1 10.1)
resources.search(aphiaid=res['worms_id'], geometry='POLYGON((30.1 10.1, 10 20, 20 40, 40 40, 30.

# Get resources for a particular eventDate
resources.search(aphiaid=res['worms_id'], year="2013", limit=20)
```

`resources.`**`resource`**(*id*, ***kwargs*)

Get resource by ID

Parameters **id** – [Fixnum] An OBIS resource identifier

Returns A dictionary

Usage:

```
from pyobis import resources
resources.resource(103)
resources.resource(2126)
```

`resources.`**`citation`**(*scientificname=None*, *aphiaid=None*, *obisid=None*, *startdate=None*, *enddate=None*, *startdepth=None*, *enddepth=None*, *geometry=None*, *year=None*, *limit=500*, *offset=0*, ***kwargs*)

List dataset citations

Parameters

- **aphiaid** – [Fixnum] A obis occurrence identifier
- **scientificname** – [String,Array] One or more scientific names from the OBIS backbone. All included and synonym taxa are included in the search.
- **year** – [Fixnum] The 4 digit year. A year of 98 will be interpreted as AD 98. Supports range queries, smaller,larger (e.g., '1990,1991', whereas '1991,1990' wouldn't work)
- **geometry** – [String] Well Known Text (WKT). A WKT shape written as either POINT, LINESTRING, LINEARRING or POLYGON. Example of a polygon: ((30.1 10.1, 20, 20 40, 40 40, 30.1 10.1)) would be queried as http://bit.ly/1BzNwDq
- **obisid** – [Fixnum] An OBIS id. This is listed as the *id* or *valid_id* in *taxa/taxon* results
- **aphiaid** – [Fixnum] An Aphia id. This is listed as the *worms_id* in *taxa/taxon* results
- **startdate** – [Fixnum] Start date
- **enddate** – [Boolean] End date
- **startdepth** – [Fixnum] Start depth
- **enddepth** – [Boolean] End depth
- **limit** – [Fixnum] Number of results to return. Default: 1000
- **offset** – [Fixnum] Start at record. Default: 0

Returns A dictionary

Usage:

```
from pyobis import resources
resources.citation(scientificname = 'Mola mola')

# Many names
resources.citation(scientificname = ['Mola', 'Abra', 'Lanice', 'Pectinaria'])

# Use paging parameters (limit and start) to page. Note the different results
# for the two queries below.
resources.citation(scientificname = 'Mola mola', offset=0, limit=10)
resources.citation(scientificname = 'Mola mola', offset=10, limit=10)
```

9.1.5 nodes module

`nodes.node`(*limit=100*, *offset=0*, ***kwargs*)

Get OBIS nodes

Parameters

- **limit** – [Fixnum] Number of results to return. Default: 1000
- **offset** – [Fixnum] Start at record. Default: 0

Returns A dictionary

Usage:

```
from pyobis import nodes
nodes.node()
nodes.node(limit = 3)
nodes.node(limit = 3, offset = 1)
```

9.1.6 checklist module

`checklist.list`(*scientificname=None*, *aphiaid=None*, *obisid=None*, *resourceid=None*, *eezid=None*, *startdate=None*, *enddate=None*, *startdepth=None*, *enddepth=None*, *geometry=None*, *year=None*, *limit=500*, *offset=0*, ***kwargs*)

Make an OBIS checklist

Parameters

- **aphiaid** – [Fixnum] A obis occurrence identifier
- **scientificname** – [String,Array] One or more scientific names from the OBIS backbone. All included and synonym taxa are included in the search.
- **year** – [Fixnum] The 4 digit year. A year of 98 will be interpreted as AD 98. Supports range queries, smaller,larger (e.g., '1990,1991', whereas '1991,1990' wouldn't work)
- **geometry** – [String] Well Known Text (WKT). A WKT shape written as either POINT, LINESTRING, LINEARRING or POLYGON. Example of a polygon: ((30.1 10.1, 20, 20 40, 40 40, 30.1 10.1)) would be queried as http://bit.ly/1BzNwDq
- **obisid** – [Fixnum] An OBIS id. This is listed as the *id* or *valid_id* in *taxa/taxon* results
- **aphiaid** – [Fixnum] An Aphia id. This is listed as the *worms_id* in *taxa/taxon* results
- **resourceid** – [Fixnum] An resource id
- **eezid** – [Fixnum] An eez id

- **startdate** – [Fixnum] Start date
- **enddate** – [Boolean] End date
- **startdepth** – [Fixnum] Start depth
- **enddepth** – [Boolean] End depth
- **limit** – [Fixnum] Number of results to return. Default: 1000
- **offset** – [Fixnum] Start at record. Default: 0

Returns A dictionary

Usage:

```
from pyobis import checklist as ch
ch.list(scientificname = 'Mola mola')

# 2005 and Cetacea
ch.list(year = 2005, scientificname = 'Cetacea')

# resourceid of 3013
ch.list(resourceid = 3013)

# eezid + scientificname
ch.list(eezid = 59, scientificname = 'Mollusca', limit = 100)

# Use paging parameters (limit and start) to page. Note the different results
# for the two queries below.
ch.list(resourceid = 3013, offset=0, limit=10)
ch.list(resourceid = 3013, offset=10, limit=10)

# Get checklist for a particular eventDate
ch.list(aphiaid=res['worms_id'], year="2013")
```

9.1.7 Changelog

0.1.0 (2016-12-12)

- first push to pypi
- finished off all OBIS API routes

0.0.6.9000 (2016-05-12)

- Updated modules with missing methods
- Added modules: groups, resources
- Removed taxon module, just a taxa module now that has all taxa/taxon methods
- Updated docs

0.0.1 (2015-12-11)

- in the works...not on pypi yet

9.2 License

MIT

9.3 Indices and tables

- genindex
- modindex
- search

p

pyobis, 22

C

citation() (pyobis.resources method), 25
common() (pyobis.taxa method), 23

D

download() (pyobis.occurrences method), 20

G

get() (pyobis.occurrences method), 20
group() (pyobis.groups method), 23

L

list() (pyobis.checklist method), 26

N

node() (pyobis.nodes method), 26

O

ObisDownload() (pyobis.occurrences method), 21

P

pyobis (module), 19, 22–24, 26

R

resource() (pyobis.resources method), 25

S

search() (pyobis.occurrences method), 19
search() (pyobis.resources method), 24
search() (pyobis.taxa method), 22

T

taxon() (pyobis.taxa method), 23
taxon_search() (pyobis.taxa method), 23