
scieloapi.py Documentation

Release 0.6

SciELO

July 23, 2014

1	User guide	3
1.1	Installation	3
1.2	Settings up the logger handler	3
1.3	Quickstart	4
2	API documentation	5
2.1	Main interface	5
	Python Module Index	9

Release v0.6. This software is licensed under [BSD License](#).

Usage example:

```
>>> import scieloapi

>>> client = scieloapi.Client('some.user', 'some.api_key')

>>> for journal in client.query('journals').all():
...     print journal['id'], journal['title']
```


Step-by-step guide to use the features provided by `scieloapi.py` for exploring data from SciELO.

1.1 Installation

This part of the documentation covers the installation process of `scieloapi.py`.

1.1.1 Pip

Installing `scieloapi.py` is simple with `pip`:

```
$ pip install scieloapi
```

1.1.2 Get the Code

`scieloapi.py` is actively developed on GitHub, check it out [here](#).

You can either clone the public repository:

```
git clone git://github.com/scieloorg/scieloapi.py.git
```

Download the `tarball`:

```
$ curl -OL https://github.com/scieloorg/scieloapi.py/tarball/master
```

Or, download the `zipball`:

```
$ curl -OL https://github.com/scieloorg/scieloapi.py/zipball/master
```

Once you have a copy of the source, you can embed it in your Python package, or install it into your site-packages easily:

```
$ python setup.py install
```

1.2 Settings up the logger handler

It is expected that the application using `scieloapi` defines a logger for `scieloapi`, e.g.:

```
logging.getLogger('scieloapi').addHandler(logging.StreamHandler())
```

See the official [docs](#) for more info.

1.3 Quickstart

When a *scieloapi.Client* instance is initialized, the process automatically introspects the API server in order to make available only the endpoints part of the specified API version. The API version may be passed as keyword argument *version* when creating the *scieloapi.Client* instance. If omitted, the highest version is used.

```
>>> client = scieloapi.Client('some.user', 'some.api_key')
```

Listing available endpoints:

```
>>> client.endpoints
[u'pressreleases', u'users', u'sections', u'sponsors', u'collections', u'changes', u'apressreleases',
>>>
```

Listing all items of an endpoint:

```
>>> for journal in client.query('journals').all(): print journal['title']
...
Acta Médica Costarricense
Acta Pediátrica Costarricense
Actualidades Investigativas en Educación
Adolescencia y Salud
Agronomía Costarricense
Agronomía Mesoamericana
Annali dell'Istituto Superiore di Sanità
Arquivos em Odontologia
Brazilian Journal of Oral Sciences
Bulletin of the World Health Organization
Cadernos de Saúde Pública
>>>
```

Listing items matching some params:

```
>>> for journal in client.query('journals').filter(collection='saude-publica'): print journal['title']
...
Annali dell'Istituto Superiore di Sanità
Bulletin of the World Health Organization
Cadernos de Saúde Pública
Ciência & Saúde Coletiva
Gaceta Sanitaria
MEDICC Review
Revista Brasileira de Epidemiologia
Revista Cubana de Salud Pública
Revista de Salud Pública
>>>
```

Getting a specific item:

```
>>> journal = client.query('journals').get(62)
>>> journal['title']
u'Acta M\xe9dica Costarricense'
>>>
```

API documentation

If you are looking for information about the library internals, this is for you.

2.1 Main interface

2.1.1 Domain-level classes

These are the classes the users will more frequently interact with.

class `scieloapi.Client` (*username*, *api_key*, *api_uri=None*, *version=None*, *connector_dep=<class 'scieloapi.core.Connector'>*, *check_ca=False*)

Collection of `Endpoint` made available in an object oriented fashion.

An instance of `Client` tries to figure out the available endpoints for the version of the API the `Client` is instantiated for, and automatically instantiates `Endpoint` for each one. If `version` is missing, the newest available will be used.

Parameters

- **username** – valid username that has access to `manager.scielo.org`.
- **api_key** – its respective api key.
- **api_uri** – (optional) if connecting to a non official instance of `SciELO Manager`
- **version** – (optional) by default the newest version is used.
- **check_ca** – (optional) if certification authority should be checked during ssl sessions. Defaults to `False`.

Usage:

```
>>> import scieloapi
>>> cli = scieloapi.Client('some.user', 'some.apikey')
<scieloapi.scieloapi.Client object at 0x10726f9d0>
>>> cli.query('journals').all()
<generator object iter_docs at 0x10fd59730>
```

endpoints

Lists all available endpoints for the api version the instance of `Client` was created to interact.

fetch_relations (*dataset*, *only=None*)

Fetches all records that relates to *dataset*.

Its important to note that only first-level relations will be fetched in order to avoid massive data retrieval.

Parameters

- **dataset** – datastructure representing a record. Typically a *dict* instance.
- **only** – (optional) a collection of relations to fetch. By default, all relations are retrieved.

Usage:

```
>>> import scieloapi
>>> cli = scieloapi.Client('some.user', 'some.apikey')
>>> cli.fetch_relations(cli.journals.get(70))
```

get (*resource_uri*)

Gets resource_uri.

Gets the given resource in a opinionated fashion in terms of the *version* passed during client's instantiation. The *endpoint* must also be available for the version the client is bound to.

Parameters **resource_uri** – text string in the form `/api/<version>/<endpoint>/<resource_id>/`.

query (*endpoint*)

Query an endpoint.

Parameters **endpoint** – string of the endpoint's name. A complete list of valid endpoints can be got at `Client.endpoints`.

version

The API version the Client instance is interfacing with.

class `scieloapi.Endpoint` (*name, connector*)

Represents an API endpoint.

Parameters

- **name** – the endpoint name.
- **connector** – instance of `Connector`.

all ()

Gets all documents of the endpoint.

filter (***kwargs*)

Gets all documents of the endpoint that satisfies some criteria.

Parameters ****kwargs** – filtering criteria as documented at docs.scielo.org

get (*resource_id*)

Gets a specific document of the endpoint.

Parameters **resource_id** – an int representing the document.

post (*data*)

Creates a new resource

Parameters **data** – serializable python data structures.

Returns id of the new resource.

2.1.2 Low-level classes and functions

Classes and functions that interface more closely with HTTP requests and responses.

class scieloapi.**Connector**(*username, api_key, api_uri=None, version=None, http_broker=None, check_ca=False*)
 Encapsulates the HTTP requests layer.

Parameters

- **username** – valid username that has access to manager.scielo.org.
- **api_key** – its respective api key.
- **api_uri** – (optional) if connecting to a non official instance of SciELO Manager
- **version** – (optional) by default the newest version is used.
- **http_broker** – (optional) a module to deal with http stuff. The reference API is implemented at `scieloapi.httpbroker`.
- **check_ca** – (optional) if certification authority should be checked during ssl sessions. Defaults to *False*.

fetch_data(*endpoint, resource_id=None, **kwargs*)
 Fetches the specified resource from the SciELO Manager API.

Parameters

- **endpoint** – a valid endpoint at <http://manager.scielo.org/api/v1/>
- **resource_id** – (optional) an int representing the document.
- ****kwargs** – (optional) params to be passed as query string.

get_endpoints()
 Get all endpoints available for the given API version.

iter_docs(*endpoint, **kwargs*)
 Iterates over all documents of a given endpoint and collection.

Parameters

- **endpoint** – must be a valid endpoint at <http://manager.scielo.org/api/v1/>
- ****kwargs** – are passed thru the request as query string params

Note that you need a valid API KEY in order to query the Manager API. Read more at: <http://ref.scielo.org/ddkpmx>

post_data(*endpoint, data*)
 Creates a new resource at *endpoint* with *data*.

Parameters

- **endpoint** – must be a valid endpoint at <http://manager.scielo.org/api/v1/>
- **data** – json serializable Python datastructures.

Returns created resource url.

scieloapi.httpbroker.**get**(**args, **kwargs*)
 Dispatches an HTTP GET request to *api_uri*.

This function is tied to some concepts of Restful interfaces like endpoints and resource ids. Any querystring params must be passed as dictionaries to *params*.

Parameters

- **api_uri** – e.g. <http://manager.scielo.org/api/v1/>
- **endpoint** – (optional) a valid endpoint at <http://manager.scielo.org/api/v1/>

- **resource_id** – (optional) an int representing the document.
- **params** – (optional) params to be passed as query string.
- **auth** – (optional) a pair of *username* and *api_key*.
- **check_ca** – (optional) if certification authority should be checked during ssl sessions. Defaults to *False*.

`scieloapi.httpbroker.post` (*api_uri*, *data*, *endpoint=None*, *auth=None*, *check_ca=False*)

Dispatches an HTTP POST request to *api_uri*, with *data*.

This function is tied to some concepts of Restful interfaces like endpoints. A new resource is created and its URL is returned.

Parameters

- **api_uri** – e.g. <http://manager.scielo.org/api/v1/>
- **data** – json serializable Python datastructures.
- **endpoint** – (optional) a valid endpoint at <http://manager.scielo.org/api/v1/>
- **auth** – (optional) a pair of *username* and *api_key*.
- **check_ca** – (optional) if certification authority should be checked during ssl sessions. Defaults to *False*.

Returns newly created resource url

S

scieloapi, 5

scieloapi.httpbroker, 7