
SciDataLib

Release v0.2.6-alpha.5

Dylan Johnson, Marshall McDonnell, Stuart Chalk

Jul 08, 2023

CONTENTS:

1	Indices and tables	9
	Python Module Index	11
	Index	13

Python library for writing SciData JSON-LD files

class scidatalib.scidata.SciData(*uid: str*)

This class is used to create and populate a SciData object, to be output as a SciData JSON-LD document

A SciData object is created by calling the SciData class i.e. SciDataObject = SciData(<uid>)

The meta variable defines the keys that make up the backbone structure of the JSON-LD document. Class methods are called to populate the meta keys

context(*context: [<class 'str'>, <class 'list'>], replace=False*) → list

Add to or replace the list of external context files

Parameters

- **context** – context URL string or list of context URL strings
- **replace** – boolean to replace or not the existing data

When called, the content URL content of the @context JSON object will be replaced or updated with the supplied list of context urls

Example:

```
SciDataObject.context(
    ['https://stuchalk.github.io/scidata/contexts/scidata.jsonld'])
```

namespaces(*namespaces: dict, replace=False*) → dict

Add to or replace the dictionary of namespaces within @context. Namespaces are needed for values in a file that reference external resources that define something (vocabulary/taxonomy/ontology entries).

Parameters

- **namespaces** – dictionary of namespaces (key->ns, val->URI start)
- **replace** – boolean to replace or not the existing data

When called, the dictionary of namespaces within the @context key of the meta variable will be replaced or updated with the supplied dictionary of namespaces

Example:

```
SciDataObject.namespaces(
    {
        "sdo": "https://stuchalk.github.io/scidata/ontology/scidata.owl#"
    }
)
```

base(*base: str*) → dict

Assign the JSON-LD @base URL (also defines '@id' under '@graph' for consistency) See: <https://www.w3.org/TR/json-ld/#base-iri>

Parameters

base – @base URL for a JSON-LD file

Defines the base url for all internal unique identifiers (defined though '@id' keyword fields). For consistency, the code also sets the '@id' field under '@graph' so that all triple subjects are unique and associated with the same graph

Example:

```
SciDataObject.graph_uid("<uniqueidentifier>")
```

docid(*docid: str*) → dict

Assign the document identifier. This will become the graph name if the file is uploaded to a graph database

Parameters

docid – the root level @id value

version(*version: str*) → dict

Assign the version of this file (not the version of the data)

Parameters

version – the top level 'version' value

graph_uid(*guid: str*) → dict

Assign the uid value within the @graph JSON object

Parameters

guid – the @graph uid value

Normally the same as the unique id used in the @graph @id value and used to easily find the data in a file system.

Example:

```
SciDataObject.graph_uid("<uniqueidentifier>")
```

author(*authors: list, replace=False*) → list

Add to or replace the list of authors within the @graph authors section

Parameters

- **authors** – list of names, or list of dicts with multiple fields
- **replace** – boolean to replace or not the existing data

Add the list of authors of a set of data with the following defined fields in the SciData context file: name, address, organization, email, orcid.

Expects either:

- 1) a list of dictionaries where each dictionary contains at minimum of a key that is 'name'

Example:

```
SciDataObject.author(  
[{'name': 'George Washington', 'ORCID': 1},  
{ 'name': 'John Adams', 'ORCID': 2}])
```

- 2) a list of strings which are author names

Example:

```
SciDataObject.author(['George Washington', 'John Adams'])
```

title(*title: str*) → str

Used to create or replace title key within @graph

Parameters

title – descriptive title of the dataset

For a data source such as a journal article, this would typically be the title of the article

Example:

```
SciDataObject.title("The Hitchhiker's Guide to the Galaxy")
```

description(*description: str*) → str

Assign the description field within @graph

Parameters

description – textual description of the dataset

Used as a brief description of the type of data. For a journal article, this might house the abstract

Example:

```
SciDataObject.description('a brief description')
```

publisher(*publisher: str*) → str

Assign the publisher field within @graph :param publisher - the name or title of the publisher of the data

This is a person, project, research group, organization etc.

Example:

```
SciDataObject.publisher('The Daily Prophet')
```

graphversion(*version: str*) → str

Assign the data version

Parameters

version – the version assigned to the data

If a version is not available, the date it was accessed online can be used to indicate the ‘state’ of the data as downloaded

Example:

```
SciDataObject.graphversion('ChEMBL database v28')
```

keywords(*keywords: [<class 'str'>, <class 'list'>], replace=False*) → list

Add to or replace the keywords of the instance

Parameters

- **keywords** – important keywords to improve data findability
- **replace** – boolean to replace or not the existing data

Example:

```
SciDataObject.keywords('important')
```

starttime(*stime: str*) → str

Assign the start time

Parameters

stime – datetime string

Typically in “%m-%d-%y %H:%M:%S” format

Example:

```
SciDataObject.starttime('04-05-21 06:14:53')
```

permalink(*link: str*) → dict

Assign the document permanent link

Parameters

link – URL to the location where this document can be found

Example:

```
SciDataObject.permalink('https://permanent.link.com/data1')
```

related(*related: [<class 'str'>, <class 'list'>], replace=False*) → list

Add to or replace the related URLs

Parameters

- **related** – URLs to other data related to this dataset
- **replace** – boolean to replace or not the existing data

Example:

```
SciDataObject.related('https://example.com/greatdata.jsonld')
```

ids(*ids: [<class 'str'>, <class 'list'>]*) → list

Add to the ids list

Parameters

ids – string or list of strings that are external references to ontological concepts

When called the contents of ‘ids’ is added to the ids list. Note that when the output function is called it iterates over instance content to find any values that are ontological references, in the format “<namespace>:<uniquevalue>”, and adds them to ids. Only ids provided in this format will be added and duplicates are ignored. Remember to add namespaces for ids.

Example:

```
SciDataObject.ids(['chebi:00001', 'qudt:GM'])
```

(requires the addition of the ‘chebi’ namespace)

discipline(*disc: str*) → str

Assign the discipline area of the data

Parameters

disc – a discipline name or identifier (preferred)

Best practice is to use an entry in an ontology, i.e. the Modern Science Ontology (<https://w3id.org/skgo/modsci#>)

Example:

```
SciDataObject.discipline('w3i:Chemistry')
```

(requires the addition of the ‘w3i’ namespace)

subdiscipline(*subdisc: str*) → str

Assign the subdiscipline area of the data

Parameters**subdisc** – a subdiscipline name or identifier (preferred)

Best practice is to use an entry in an ontology, i.e. the Modern Science Ontology (<https://w3id.org/skgo/modsci#>)

Example:

```
SciDataObject.subdiscipline('w3i:AnalyticalChemistry')
```

evaluation(*evaln: str*) → str

Assign the evaluation field

Parameters**evaln** – the method of evaluation of research data

Recommended values of this field are: experimental, theoretical, computational

Example:

```
SciDataObject.evaluation('experimental')
```

aspects(*aspects: list*) → list

Add to or replace the aspects of the file

Example:

```
SciDataObject.aspects([{"@id": "assay",
"@type": "sdo:assay",
"description": "Inhibition of human ERG "
               "by MK499 binding assay",
"assay_organism": "Homo sapiens"}])
```

Method also accepts keyword '#intlinks'. See documentation for def scidatapackage.

facets(*facets: list*) → list

Add to or replace the facets of the file

Example:

```
SciDataObject.facets([{"@id": "compound",
"@type": "sdo:compound",
"mw_freebase": "491.52",
"full_molformula": "C26H26FN5O4"}])
```

Method also accepts keyword '#intlinks'. See documentation for def scidatapackage.

scope(*scope: [<class 'str'>, <class 'list'>]*) → str

Assign what thing(s) the dataset relates to

Parameters**scope** – str or list of internal unique id(s) of entity(ies) in the system to which the data describes

The scope of a datasets should be described in the 'system' 'facets' section, e.g. chemical system, organism, specimen, should be included as a scope using the defined unique '@id' for that section

Example:

```
SciDataObject.scope('chemicalsystem/1/')
```

attribute(*attributes: list*) → list

Add one or more attributes

datapoint(*points: list*) → list

Add one or more datapoints

Example:

```
SciDataObject.datapoint(
  [{"@id": "datapoint",
    "@type": "sdo:datapoint",
    "data": [{"@id": "datum",
      "@type": "sdo:exptdata",
      "type": "IC50",
      "value": "15.2",
      "units": "uM"}]}])
```

dataseries(*series: list*) → list

Add one or more dataseries

datagroup(*group: list*) → list

Add one or more datagroups

scidatapackage(*package*)

Add a package of data where the datapoints are linked with the associated aspects and facets. A package contains one or more ‘packets’ of associated aspects, facets and datapoints.

Template:

```
package = [
  {'aspects': {}, 'facets': {}, 'datapoints': {}},
  {'aspects': {}, 'facets': {}, 'datapoints': {}},
]
```

Example:

```
SciDataObject.scidatapackage([
  {
    "aspects": [
      {
        "@id": "assay/",
        "@type": "sdo:assay",
        "description": "Inhibition of human ERG by MK499 binding assay",
        "assay_organism": "Homo sapiens"
      }
    ],
    "facets": [
      {
        "@id": "compound/",
        "@type": "sdo:compound",
        "mw_freebase": "491.52",
        "full_molformula": "C26H26FN5O4",
        "#intlinks": [
          {
            "@id": "identifier/",
            "@type": "sdo:identifier",
            "standard_inchi_key": "O1NHUVBCKUJZAG-UHFFFAOYSA-N"
          }
        ]
      }
    ]
  }
])
```

(continues on next page)

(continued from previous page)

```

    }}
  },
  {
    "@id": "target/",
    "@type": "sdo:target",
    "pref_name": "HERG",
    "tax_id": 9606,
    "organism": "Homo sapiens"
  }
],
"datapoints": [{
  "@id": "datapoint/",
  "@type": "sdo:datapoint",
  "data": [{
    "@id": "datum",
    "@type": "sdo:exptdata",
    "type": "IC50",
    "value": "15.2",
    "units": "uM"
  }]
}]
}]
})

```

sources(*sources: list, replace=False*) → dict

Add to or replace the source reference list

Parameters

- **sources** (*list*) – information about where the data came from
- **replace** (*bool (default: False)*) – replace (True) or add to the existing sources (False)

Add a list of sources with any of the available defined fields in the SciData context file: citation, reftype, url, doi

Example:

```

SciDataObject.sources([
{'citation': 'Chalk, S.J. SciData: a data model and
ontology for semantic representation of scientific data.
J Cheminform 8, 54 (2016)',
doi': 'https://doi.org/10.1186/s13321-016-0168-9'})

```

rights(*holder: str, lic: str*) → dict

Add the rights section to the file (max: 1 entry)

Parameters

- **holder** – the entity that holds the license to this data
- **lic** – the assigned license

property output: dict

Completes and cleans a Scidata Object (instance of this class) before its output.

INDICES AND TABLES

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

PYTHON MODULE INDEX

S

scidatalib.scidata, ??

INDEX

A

aspects() (*scidatalib.scidata.SciData method*), 5
attribute() (*scidatalib.scidata.SciData method*), 6
author() (*scidatalib.scidata.SciData method*), 2

B

base() (*scidatalib.scidata.SciData method*), 1

C

context() (*scidatalib.scidata.SciData method*), 1

D

datagroup() (*scidatalib.scidata.SciData method*), 6
datapoint() (*scidatalib.scidata.SciData method*), 6
dataseries() (*scidatalib.scidata.SciData method*), 6
description() (*scidatalib.scidata.SciData method*), 3
discipline() (*scidatalib.scidata.SciData method*), 4
docid() (*scidatalib.scidata.SciData method*), 2

E

evaluation() (*scidatalib.scidata.SciData method*), 5

F

facets() (*scidatalib.scidata.SciData method*), 5

G

graph_uid() (*scidatalib.scidata.SciData method*), 2
graphversion() (*scidatalib.scidata.SciData method*), 3

I

ids() (*scidatalib.scidata.SciData method*), 4

K

keywords() (*scidatalib.scidata.SciData method*), 3

M

module
 scidatalib.scidata, 1

N

namespaces() (*scidatalib.scidata.SciData method*), 1

O

output (*scidatalib.scidata.SciData property*), 7

P

permalink() (*scidatalib.scidata.SciData method*), 4
publisher() (*scidatalib.scidata.SciData method*), 3

R

related() (*scidatalib.scidata.SciData method*), 4
rights() (*scidatalib.scidata.SciData method*), 7

S

SciData (*class in scidatalib.scidata*), 1
scidatalib.scidata
 module, 1
scidatapackage() (*scidatalib.scidata.SciData method*), 6
scope() (*scidatalib.scidata.SciData method*), 5
sources() (*scidatalib.scidata.SciData method*), 7
starttime() (*scidatalib.scidata.SciData method*), 3
subdiscipline() (*scidatalib.scidata.SciData method*), 2
 4

T

title() (*scidatalib.scidata.SciData method*), 2

V

version() (*scidatalib.scidata.SciData method*), 2