
SciBib

Release 0.1.0

Gaël Cousin

Jan 14, 2024

CONTENTS

1	Indices and tables	1
1.1	The SciBib Package	1
1.2	The data_query module	1
1.3	The abstract_collector module	4
	Python Module Index	5
	Index	7

INDICES AND TABLES

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

1.1 The SciBib Package

This package enables scientific bibliographical data retrieval from an author's Orcid id. The main goal is to collect bibtex entries for the authors works and to collect abstracts for theses works.

Bibtex collection works fine provided the author has an orcid record and the sought article is referenced there, see `data_query.OrcidWork.bibtex`.

Abstract retrieval can be performed using ArXiv's API if the article is on the Arxiv and the author associated her/his orcid id with her/his arxiv account, see `data_query.AuthorData.work_summary_from_arxiv`.

Another option is to get a (sometimes more up-to-date) abstract scraping the journal's website. In this case, some legal or technical obstructions might appear. However, a tool to try this technique is provided, namely the `scrape_abstract` method of our `OrcidWork` class.

Another useful feature is to use the doi of a work to build an url that leads to the article in the publisher's website. This can be obtained with `OrcidWork.doi`.

Other data sources and other outputs could be added in the future, depending of the users' suggestions/pull requests.

1.2 The `data_query` module

This module defines two classes that allow to parse author data from Orcid and arxiv. These are `AuthorData` and `OrcidWork`.

`class scibib.data_query.OrcidWork`

Methods:

<code>__init__(work_data)</code>	Instantiate single work object.
<code>scrape_abstract()</code>	Scrape the work's summary from the editor/journal's site.

Attributes:

<code>path</code>	Orcid path to the data.
<code>title</code>	Work title.
<code>doi</code>	The Work's doi.
<code>bibtex</code>	Return the bibtex entry for self from source.

`__init__`(*work_data*)

Instantiate single work object.

Parameters

`work_data` (*nested lists/dictionaries*) – part of a loaded json data corresponding to a single work, as obtained from orcid's API.

property `path`

Orcid path to the data.

property `title`

Work title.

property `doi`

The Work's doi.

Returns

the doi.

Return type

str

property `bibtex`

Return the bibtex entry for self from source.

Parameters

- **`source`** (*str, optional*) – Equals 'doi'. Defaults to 'doi'.
- **`future`**. (*Other sources might be available in the*) –

`scrape_abstract`()

Scrape the work's summary from the editor/journal's site. Beware that you might need authorization from the editor/journal to use this functionality.

class `scibib.data_query.AuthorData`

A class to parse Orcid author entries.

Methods:

<code>__init__</code> (<i>orcid_id</i>)	Instantiator
<code>work_summary_from_arxiv</code> (<i>orcid_work</i>)	Match work with an arxiv entry to provide a summary.

Attributes:

<i>orcid_record</i>	The raw orcid record as a parsed json.
<i>arxiv_record</i>	The raw arxiv record as an atom feed.
<i>articles</i>	list of article entries in the author's Orcid entry.
<i>orcid_id_is_on_arxiv</i>	Check if the author associated his/her Arxiv with Orcid.
<i>arxiv_summaries_dic</i>	Return dict that maps arxiv_entries -> abstracts for the author.

__init__(*orcid_id*)

Instantiator

Parameters

orcid_id(*str*) – The author's orcid id

property orcid_record

The raw orcid record as a parsed json.

Returns

The raw orcid record as a parsed json (using json.load).

Return type

list

property arxiv_record

The raw arxiv record as an atom feed.

property articles

list of article entries in the author's Orcid entry.

Returns

list of article entries, formatted as OrcidWork instances.

Return type

list[*OrcidWork*]

property orcid_id_is_on_arxiv

Check if the author associated his/her Arxiv with Orcid.

Returns

True if yes, False if no!

Return type

bool

property arxiv_summaries_dic

Return dict that maps arxiv_entries -> abstracts for the author.

work_summary_from_arxiv(*orcid_work*)

Match work with an arxiv entry to provide a summary.

Parameters

orcid_work(*OrcidWork*) – the work that needs summary.

Returns

The guessed summary

Return type

str

1.3 The `abstract_collector` module

This module defines the `main_paragraph` function.

Functions:

<code>main_paragraph(url)</code>	From a web page, return the paragraph with the biggest length.
----------------------------------	--

`scibib.abstract_collector.main_paragraph(url)`

From a web page, return the paragraph with the biggest length.

Parameters

url (*str*) – the url of the web page to treat.

PYTHON MODULE INDEX

S

`scibib`, [1](#)
`scibib.abstract_collector`, [3](#)
`scibib.data_query`, [1](#)

Symbols

`__init__()` (*scibib.data_query.AuthorData* method), 3
`__init__()` (*scibib.data_query.OrcidWork* method), 2

A

`articles` (*scibib.data_query.AuthorData* property), 3
`arxiv_record` (*scibib.data_query.AuthorData* property), 3
`arxiv_summaries_dic` (*scibib.data_query.AuthorData* property), 3
`AuthorData` (class in *scibib.data_query*), 2

B

`bibtex` (*scibib.data_query.OrcidWork* property), 2

D

`doi` (*scibib.data_query.OrcidWork* property), 2

M

`main_paragraph()` (in *scibib.abstract_collector*), 4
module
 scibib, 1
 scibib.abstract_collector, 3
 scibib.data_query, 1

O

`orcid_id_is_on_arxiv` (*scibib.data_query.AuthorData* property), 3
`orcid_record` (*scibib.data_query.AuthorData* property), 3
`OrcidWork` (class in *scibib.data_query*), 1

P

`path` (*scibib.data_query.OrcidWork* property), 2

S

scibib
 module, 1
scibib.abstract_collector

module, 3
scibib.data_query
 module, 1
`scrape_abstract()` (*scibib.data_query.OrcidWork* method), 2

T

`title` (*scibib.data_query.OrcidWork* property), 2

W

`work_summary_from_arxiv()` (*scibib.data_query.AuthorData* method), 3