
python-doi

Release 0.1.1

Alejandro Gallo

Dec 29, 2019

CONTENTS:

1	python-doi	1
1.1	Installation	1
1.2	Usage	1
2	Contributing	3
2.1	Types of Contributions	3
2.2	Get Started!	4
2.3	Pull Request Guidelines	4
3	Credits	5
3.1	Development Lead	5
3.2	Contributors	5
4	API	7
5	Indices and tables	9
	Python Module Index	11
	Index	13

PYTHON-DOI

Python package to work with Document Object Identifier (doi)

Development of python-doi happens on [Github](#).

1.1 Installation

To install the latest released version of python-doi, run this command in your terminal:

```
$ pip install python-doi
```

This is the preferred method to install python-doi, as it will always install the most recent stable release.

If you don't have [pip](#) installed, this [Python installation guide](#) can guide you through the process.

To install the latest development version of python-doi from [Github](#).

```
$ pip install git+https://github.com/papis/python-doi.git@master#egg=doi
```

1.2 Usage

To use python-doi in a project:

```
import doi
```


CONTRIBUTING

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

2.1 Types of Contributions

2.1.1 Report Bugs

Report bugs at <https://github.com/papis/python-doi/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

2.1.2 Fix Bugs / Implement Features

Look through the GitHub issues for bugs or feature requests. Anybody is welcome to submit a pull request for open issues.

2.1.3 Write Documentation

python-doi could always use more documentation, whether as part of the official python-doi docs, in docstrings, or even on the web in blog posts, articles, and such.

2.1.4 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/papis/python-doi/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

2.2 Get Started!

In short,

1. Clone the repository from `git@github.com:papis/python-doi.git`
2. Fork the repo on GitHub to your personal account.
3. Add your fork as a remote.
4. Pull in the latest changes from the master branch.
5. Create a topic branch
6. Make your changes and commit them (testing locally)
7. Push changes to the topic branch on *your* remote
8. Make a pull request against the base master branch through the Github website of your fork.

The project contains a `Makefile` to help with development tasks. In your checked-out clone, do

```
$ make help
```

to see the available make targets.

2.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in `README.rst`.
3. Check https://travis-ci.org/papis/python-doi/pull_requests and make sure that the tests pass for all supported Python versions.

CREDITS

3.1 Development Lead

- Alejandro Gallo <aamsgallo@gmail.com>

3.2 Contributors

None yet. Why not be the first?

`doi.pdf_to_doi(filepath, maxlines=None)`

Try to get DOI from a filepath. It looks for a regex in the binary data and returns the first DOI found, in the hopes that this DOI is the correct one.

Parameters

- **filepath** (`str`) – Path to the pdf file.
- **maxlines** (`Optional[int]`) – Maximum number of lines that should be checked For some documents, it could spend a long time trying to look for a DOI, and DOIs in the middle of documents don't tend to be the correct DOI of the document.

Return type `Optional[str]`

Returns DOI or None.

`doi.validate_doi(doi)`

We check that the DOI can be resolved by [official means](#). If so, we return the resolved URL, otherwise, we return None (which means the DOI is invalid).

Parameters **doi** (`str`) – Identifier.

Return type `Optional[str]`

Returns The URL assigned to the DOI or None.

`doi.get_clean_doi(doi)`

Check if the DOI is actually a URL and in that case just get the exact DOI.

Parameters **doi** (`str`) – String containing a DOI.

Return type `str`

Returns The extracted DOI.

`doi.find_doi_in_text(text)`

Try to find a DOI in a text.

Parameters **text** (`str`) – Text in which to look for DOI.

Return type `Optional[str]`

Returns A DOI, if found, otherwise None.

`doi.get_real_url_from_doi(doi)`

Get a URL corresponding to a DOI.

Parameters **doi** (`str`) – Identifier.

Return type `Optional[str]`

Returns A URL for the DOI. If the DOI is invalid, return `None`.

INDICES AND TABLES

- `genindex`
- `modindex`

PYTHON MODULE INDEX

d

doi, [7](#)

INDEX

D

`doi` (*module*), 7

F

`find_doi_in_text()` (*in module doi*), 7

G

`get_clean_doi()` (*in module doi*), 7

`get_real_url_from_doi()` (*in module doi*), 7

P

`pdf_to_doi()` (*in module doi*), 7

V

`validate_doi()` (*in module doi*), 7