
searchdocs

Release 0.2.2

Search the Python documentation from your terminal.

Dominic Davis-Foster

May 15, 2024

Contents

1	Installation	3
1.1	from PyPI	3
1.2	from GitHub	3
2	Usage	5
2.1	searchdocs	5
3	Public API	7
3.1	cache_dir	7
3.2	cache_dir_for_url	7
3.3	download_objects_inv	7
3.4	find_url	8
3.5	resolve_url	8
4	Contributing	9
4.1	Coding style	9
4.2	Automated tests	9
4.3	Type Annotations	9
4.4	Build documentation locally	10
5	Downloading source code	11
5.1	Building from source	12
6	License	13
	Python Module Index	15
	Index	17

Built on top of `sphobjinv`, which can be used for more advanced manipulation and searching of Sphinx `objects.inv` files.

Installation

1.1 from PyPI

```
$ python3 -m pip install searchdocs --user
```

1.2 from GitHub

```
$ python3 -m pip install git+https://github.com/domdfcoding/searchdocs@master --user
```


Usage

2.1 searchdocs

Search for `SEARCH_TERM` in the Python documentation, and print the URL of the best match.

```
searchdocs [OPTIONS] SEARCH_TERM
```

Options

--browser

Open the documentation in the default web browser.

Arguments

SEARCH_TERM

Required argument.

Tip: The output from the `searchdocs` command can be piped into the `lynx (1)` command to open a web browser in the terminal:

```
$ searchdocs rmtree | lynx -
```

Public API

Search the Python documentation from your terminal.

Data:

<code>cache_dir</code>	Directory in which cached files are stored.
------------------------	---

Functions:

<code>cache_dir_for_url(url)</code>	Returns the path to the cache subdirectory for the given URL.
<code>download_objects_inv(docs_url)</code>	Download the Sphinx <code>objects.inv</code> file for the documentation available at the given URL.
<code>find_url(docs_url, search_term)</code>	Find the complete documentation URL for the given function, class, method etc.
<code>resolve_url(url)</code>	Resolve any redirects in the given URL.

```
cache_dir = PosixPathPlus('/home/docs/.cache/searchdocs')
```

Type: `PosixPathPlus`

Directory in which cached files are stored.

```
cache_dir_for_url(url)
```

Returns the path to the cache subdirectory for the given URL.

Parameters `url` (`Union[str, URL]`)

Return type `PathPlus`

```
download_objects_inv(docs_url)
```

Download the Sphinx `objects.inv` file for the documentation available at the given URL.

Parameters `docs_url` (`Union[str, RequestsURL]`) – The base URL for the documentation, e.g. `"https://docs.python.org/3/"`.

Return type `PathPlus`

Returns The filename of the cached file.

find_url (*docs_url*, *search_term*)

Find the complete documentation URL for the given function, class, method etc.

Parameters

- **docs_url** (`Union[str, RequestsURL]`) – The base URL for the documentation, e.g. "https://docs.python.org/3/".
- **search_term** (`str`) – The object to search for, e.g. 'TemporaryDirectory'.

Return type `URL`

Returns The url of the object in the documentation, e.g. `URL('https://docs.python.org/3/library/tempfile.html#tempfile.TemporaryDirectory')`.

resolve_url (*url*)

Resolve any redirects in the given URL.

Parameters **url** (`Union[str, RequestsURL]`)

Return type `RequestsURL`

Contributing

searchdocs uses `tox` to automate testing and packaging, and `pre-commit` to maintain code quality.

Install `pre-commit` with `pip` and install the git hook:

```
$ python -m pip install pre-commit
$ pre-commit install
```

4.1 Coding style

`formate` is used for code formatting.

It can be run manually via `pre-commit`:

```
$ pre-commit run formate -a
```

Or, to run the complete autoformatting suite:

```
$ pre-commit run -a
```

4.2 Automated tests

Tests are run with `tox` and `pytest`. To run tests for a specific Python version, such as Python 3.6:

```
$ tox -e py36
```

To run tests for all Python versions, simply run:

```
$ tox
```

4.3 Type Annotations

Type annotations are checked using `mypy`. Run `mypy` using `tox`:

```
$ tox -e mypy
```

4.4 Build documentation locally

The documentation is powered by Sphinx. A local copy of the documentation can be built with `tox`:

```
$ tox -e docs
```

Downloading source code

The searchdocs source code is available on GitHub, and can be accessed from the following URL: <https://github.com/domdfcoding/searchdocs>

If you have git installed, you can clone the repository with the following command:

```
$ git clone https://github.com/domdfcoding/searchdocs
```

```
Cloning into 'searchdocs'...
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (41/41), done.
remote: Total 173 (delta 16), reused 17 (delta 6), pack-reused 126
Receiving objects: 100% (173/173), 126.56 KiB | 678.00 KiB/s, done.
Resolving deltas: 100% (66/66), done.
```

Alternatively, the code can be downloaded in a 'zip' file by clicking:

Clone or download → Download Zip

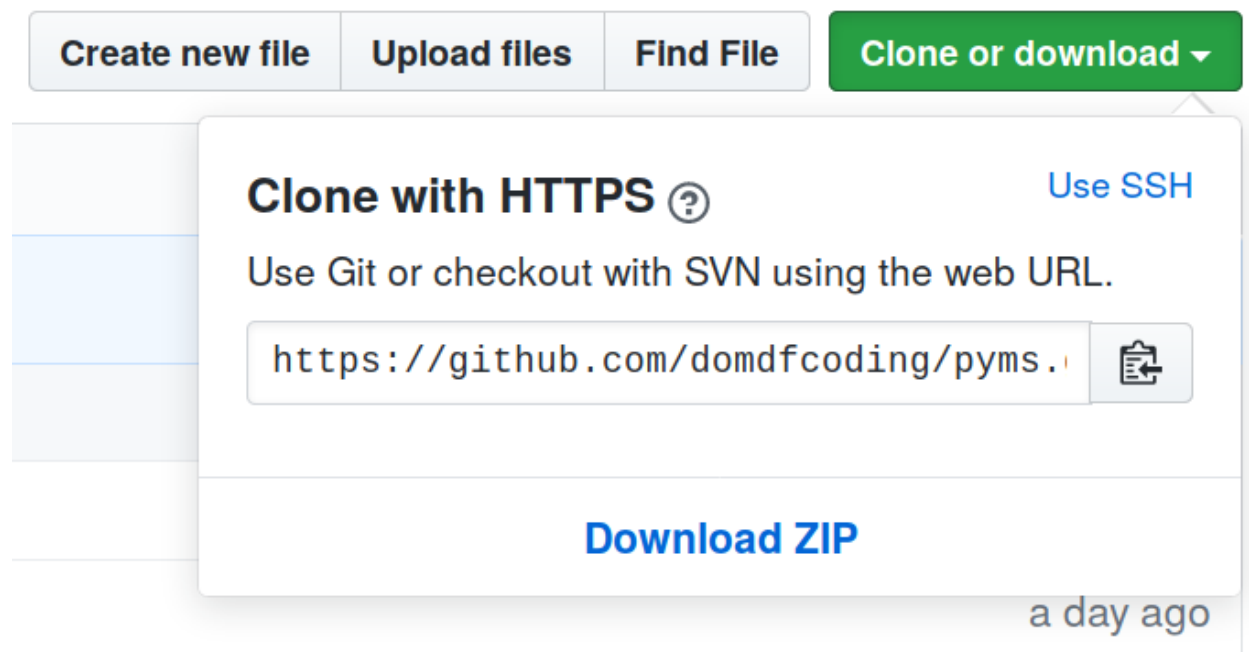


Fig. 1: Downloading a 'zip' file of the source code

5.1 Building from source

The recommended way to build `searchdocs` is to use `tox`:

```
$ tox -e build
```

The source and wheel distributions will be in the directory `dist`.

If you wish, you may also use `pep517.build` or another **PEP 517**-compatible build tool.

License

searchdocs is licensed under the [MIT License](#)

A short and simple permissive license with conditions only requiring preservation of copyright and license notices. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

Permissions

- Commercial use – The licensed material and derivatives may be used for commercial purposes.
- Modification – The licensed material may be modified.
- Distribution – The licensed material may be distributed.
- Private use – The licensed material may be used and modified in private.

Conditions

- License and copyright notice – A copy of the license and copyright notice must be included with the licensed material.

Limitations

- Liability – This license includes a limitation of liability.
- Warranty – This license explicitly states that it does NOT provide any warranty.

[See more information on choosealicense.com](#) ⇒

```
Copyright (c) 2021 Dominic Davis-Foster
```

```
Permission is hereby granted, free of charge, to any person obtaining a copy  
of this software and associated documentation files (the "Software"), to deal  
in the Software without restriction, including without limitation the rights  
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
copies of the Software, and to permit persons to whom the Software is  
furnished to do so, subject to the following conditions:
```

```
The above copyright notice and this permission notice shall be included in all  
copies or substantial portions of the Software.
```

```
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,  
EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF  
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.  
IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,  
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR  
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE  
OR OTHER DEALINGS IN THE SOFTWARE.
```


Python Module Index

S

`searchdocs`, [7](#)

Symbols

`--browser`
searchdocs command line option, [5](#)

C

`cache_dir` (*in module searchdocs*), [7](#)
`cache_dir_for_url()` (*in module searchdocs*), [7](#)

D

`download_objects_inv()` (*in module searchdocs*), [7](#)

F

`find_url()` (*in module searchdocs*), [8](#)

M

MIT License, [13](#)
module
searchdocs, [7](#)

P

Python Enhancement Proposals
PEP 517, [12](#)

R

`resolve_url()` (*in module searchdocs*), [8](#)

S

`SEARCH_TERM`
searchdocs command line option, [5](#)
searchdocs
module, [7](#)
searchdocs command line option
`--browser`, [5](#)
`SEARCH_TERM`, [5](#)