pywebdoc

Quentin Deschamps

Jan 30, 2024

CONTENTS:

CHAPTER

ONE

PRESENTATION

pywebdoc is a CLI application to quickly open web documentation about Python. This tool works for standard libraries and PyPI packages. It opens web pages on your default web browser.

1.1 Installation

You can install latest release of *pywebdoc* using `pip3`:

```
pip3 install pywebdoc
```

1.2 CLI Reference

The command line interface of *pywebdoc* is available for Linux, macOS and Windows. It uses *click*. To check if you successfully installed the library, you can entry in a command prompt:

```
pywebdoc --help
```

You will see the following content:

```
Usage: pywebdoc [OPTIONS] COMMAND [ARGS]...

Open Python packages url.

Options:
--version      Show the version and exit.
-v, --verbose  Give more output
--help         Show this message and exit.

Commands:
home           Open the home-page of a PyPI package.
list-packages  Make HTML file with the list of installed PyPI packages.
list-std       List standard libraries documentation urls.
py             Open the Python official documentation.
pypi           Open the PyPI web page of a package.
rtd            Open the documentation page of a package on ReadTheDocs.
std            Open the documentation page of a standard Python library.
```

It is the help menu of the CLI. You can see all the commands you can use.

1.2.1 Python official documentation

To **open the Python official documentation**, use the `py` command:

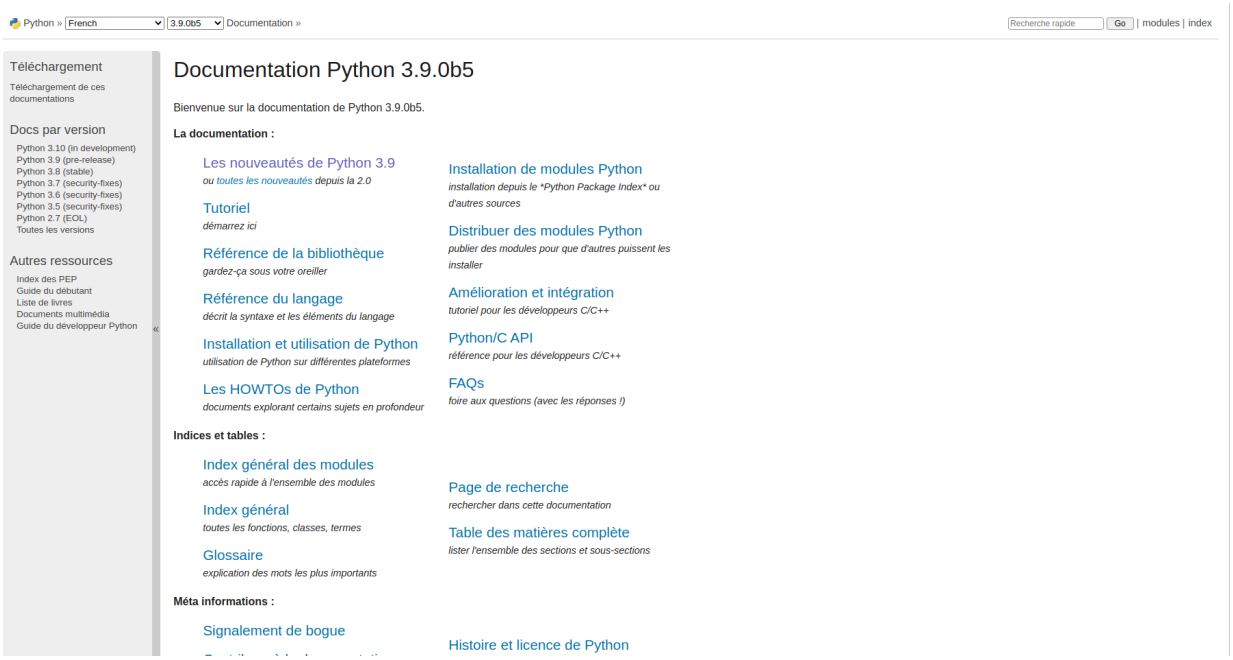
```
pywebdoc py [OPTIONS]
```

You can set the Python version with the VERSION option (default: 3) and the language with the LANG option (default: english).

For example, to open Python 3.9 documentation in french:

```
pywebdoc py -v 3.9 -l fr
```

This command will open your default web browser on this page:



1.2.2 Standard library

To **open the documentation page of a standard Python library**, use the `std` command:

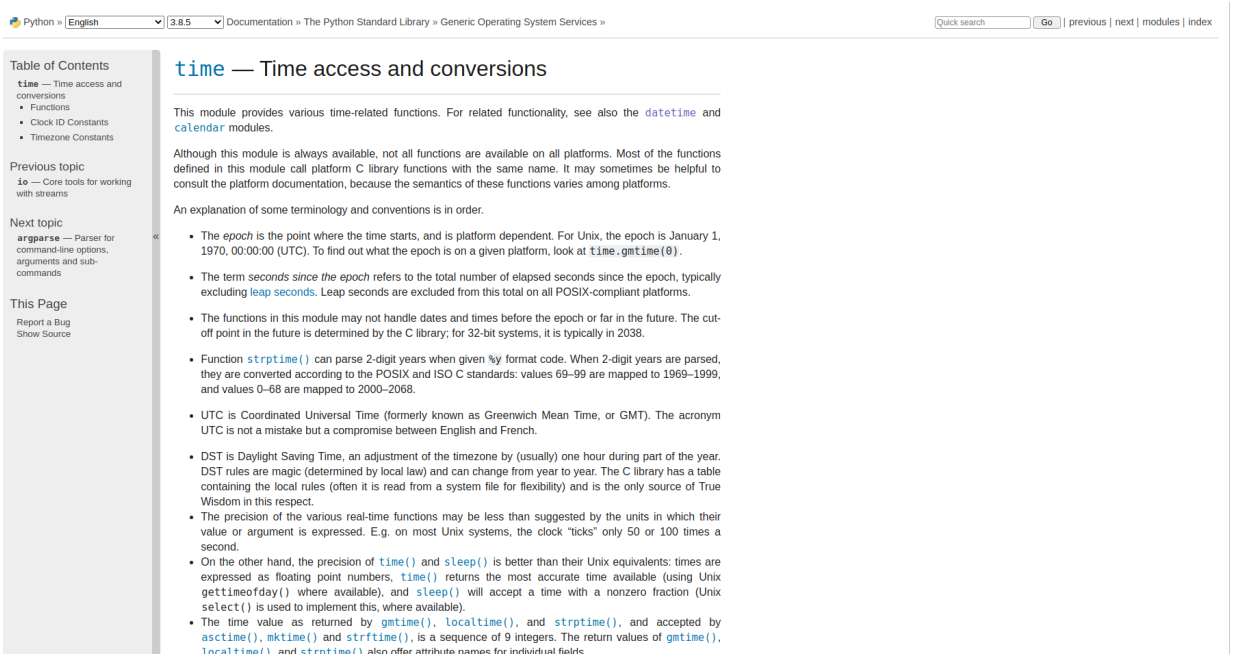
```
pywebdoc std [OPTIONS] LIBRARY
```

The LIBRARY argument is the name of the library or module. You can also set the Python version and the language using the same options than the `py` command.

For example, to open the documentation of *time*:

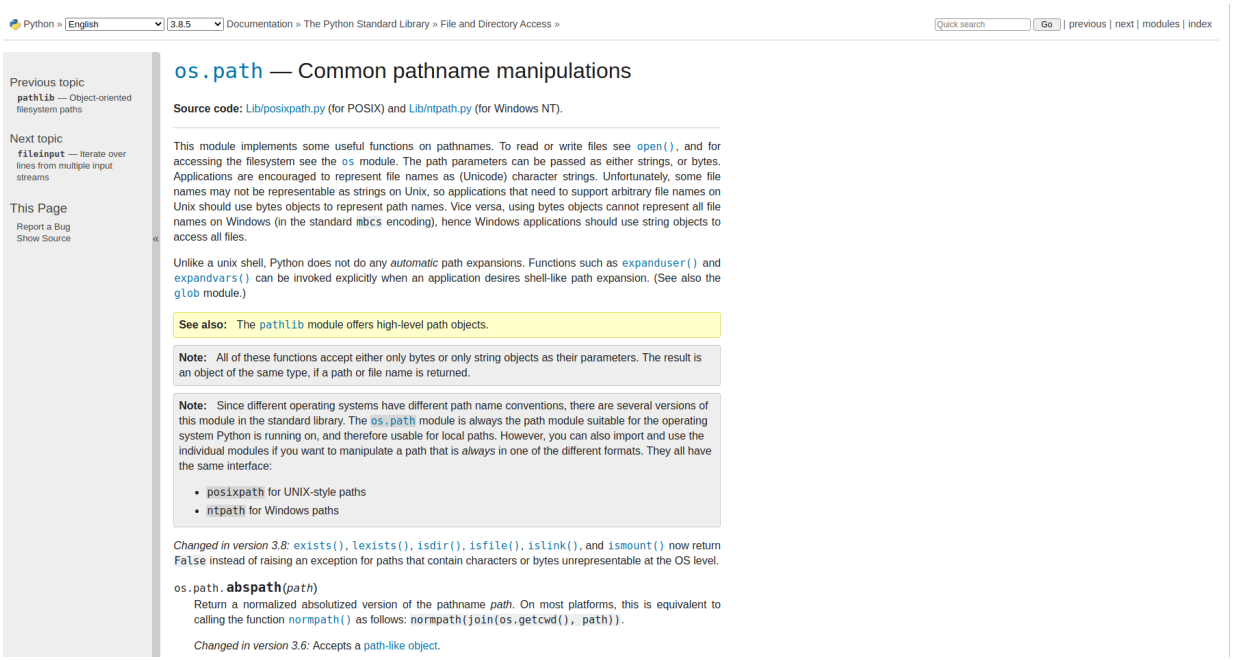
```
pywebdoc std time
```

This is the web page opened:



You can also open modules like *os.path*:

```
pywebdoc std os.path
```



Note: If the URL does not exist, you see a "**404 Not Found**" error in the command prompt.

1.2.3 PyPI package

Pywebdoc can also open URLS about **PyPI packages**.

PyPI web page

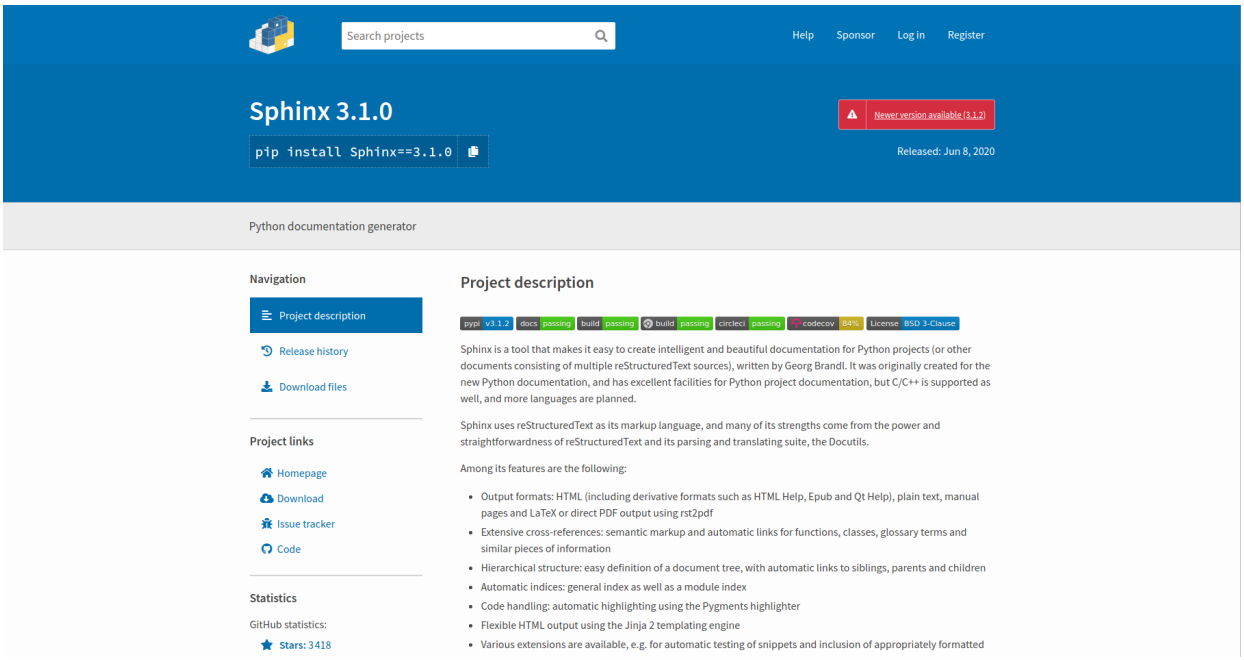
First, you can **open the PyPI web page of a package** using the `pypi` command:

```
pywebdoc pypi [OPTIONS] PACKAGE
```

The PACKAGE argument is the name of the package. You can choose the release version with the VERSION option. For example, to open the PyPI web page of *Sphinx* 3.1.0:

```
pywebdoc pypi sphinx -v 3.1.0
```

This command will open this web page:



Home-page

You can also **open the home-page of a PyPI package** using the `home` command:

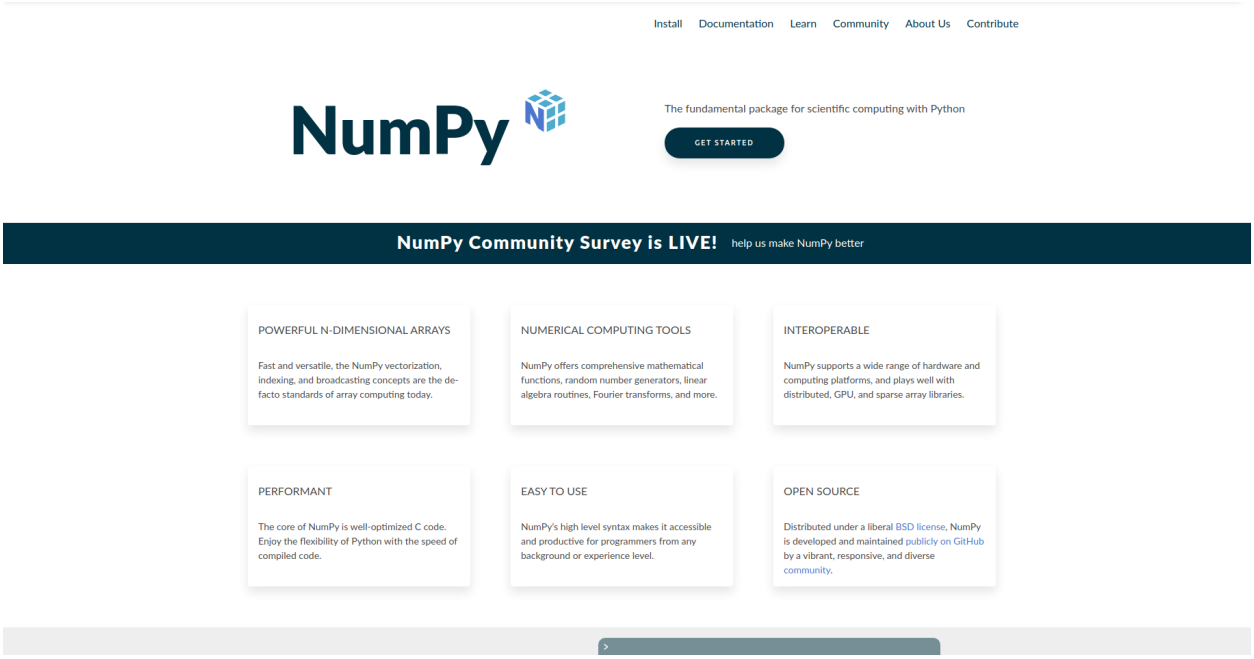
```
pywebdoc home [OPTIONS] PACKAGE
```

> **Warning:** This command will call the `pip` command to get the URL of the home-page. So, **the package must be installed**.

For example, to open the home-page of *numpy*, use:

```
pywebdoc home numpy
```

This command will open this URL:



ReadTheDocs documentation

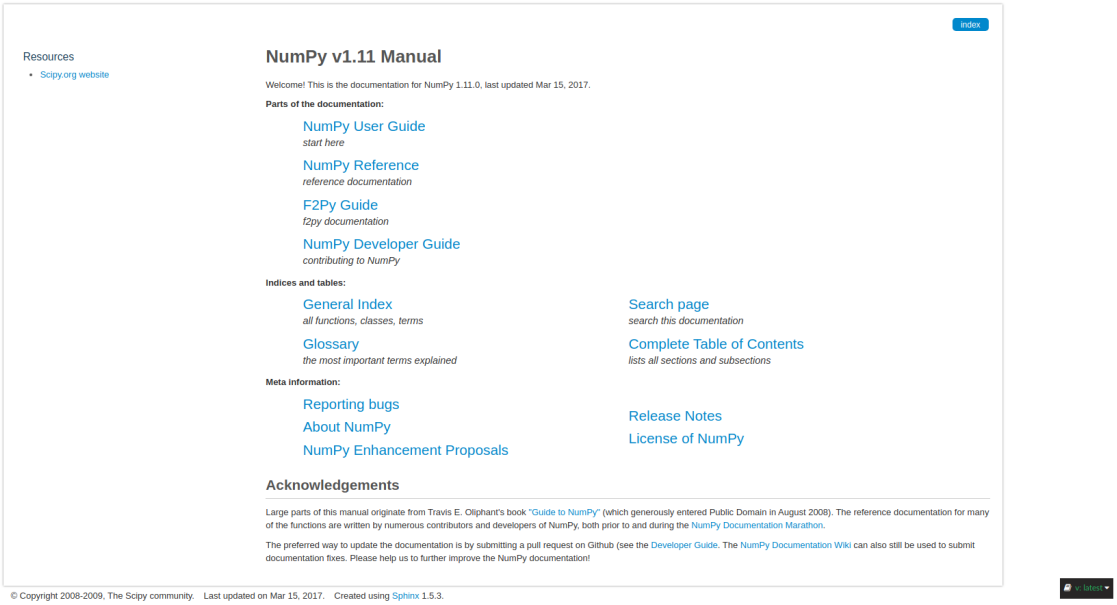
Finally, some PyPI packages host documentation on ReadTheDocs. To **open the documentation page of a package on ReadTheDocs**, use the command:

```
pywebdoc rtd [OPTIONS] PACKAGE
```

You can choose the documentation version with the VERSION option (default: latest) and the language with the LANG option (default: en). For example, you can open the documentation of *numpy* using:

```
pywebdoc rtd numpy
```

None



1.2.4 Listing

With *pywebdoc*, you can also see the list of your packages and libraries and their documentation.

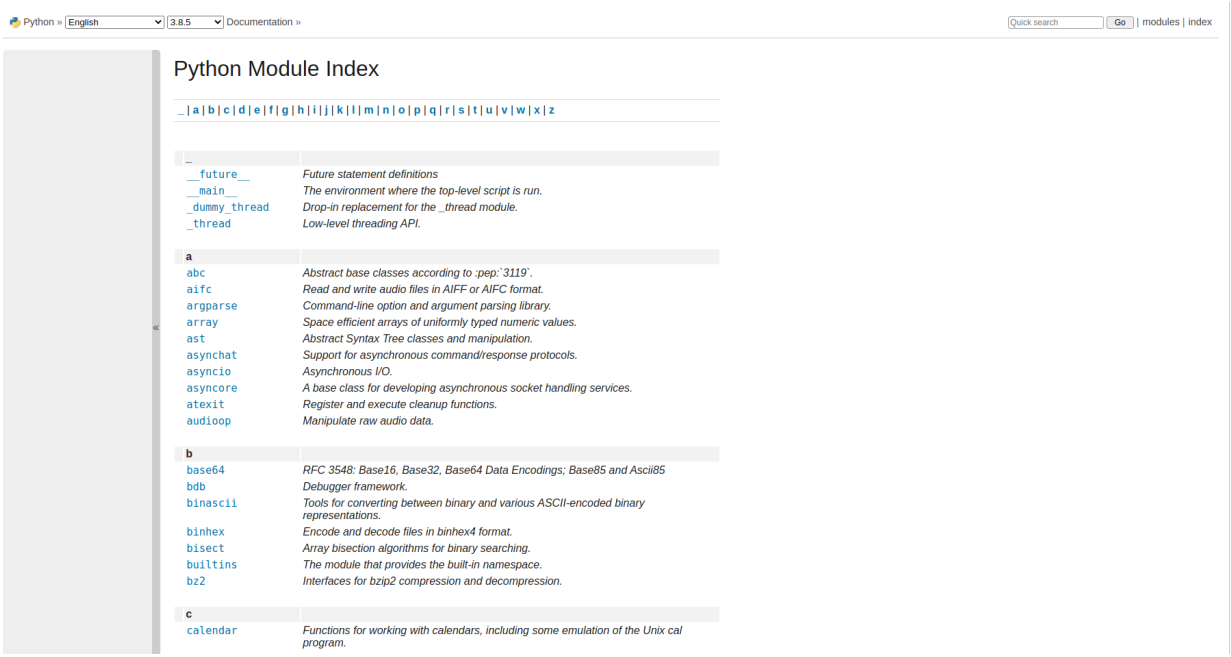
- To get the list of **standard Python libraries**, use the `list-std` command:

```
pywebdoc list-std [OPTIONS]
```

You can use the same options as the `py` command. For example, the command

```
pywebdoc list-std
```

will open this URL:



- To get the list of **installed PyPI packages**, use the `list-packages` command:

```
pywebdoc list-packages [OPTIONS]
```

This command will create a HTML file with the list of installed PyPI packages. The creation of this page may take several minutes. On this page, you will see the following informations about packages:

- name
- version
- summary
- home-page
- license

Once the file is created, it will be opened immediately. If you need to update the HTML file, use the UPDATE option.

For example, you can see below an example of this file:

PyPI installed packages

Package	Version	Summary	Home-page	License
affapy	0.1	A Python library for multiprecision Affine Arithmetic	https://gitlab.lip6.fr/hilaire/affapy	UNKNOWN
alabaster	0.7.12	A configurable sidebar-enabled Sphinx theme	https://alabaster.readthedocs.io	UNKNOWN
appdirs	1.4.3	A small Python module for determining appropriate platform-specific dirs, e.g. a "user data dir".	http://github.com/ActiveState/appdirs	MIT
apturl	0.5.2	UNKNOWN		UNKNOWN
astroid	2.3.3	An abstract syntax tree for Python with inference support.	https://github.com/PyCQA/astroid	LGPL
attrs	19.3.0	Classes Without Boilerplate	https://www.attrs.org/	MIT
Babel	2.8.0	Internationalization utilities	http://babel.pocoo.org/	BSD
backcall	0.1.0	Specifications for callback functions passed in to an API	https://github.com/takluyver/backcall	BSD
bcrypt	3.1.7	Modern password hashing for your software and your servers	https://github.com/pyca/bcrypt/	Apache License, Version 2.0
beautifulsoup4	4.9.1	Screen-scraping library	http://www.crummy.com/software/BeautifulSoup/bs4/	MIT
binary	1.0.0	UNKNOWN	https://github.com/ofek/binary	MIT/Apache-2.0
bleach	3.1.4	An easy safelist-based HTML-sanitizing tool.	https://github.com/mozilla/bleach	Apache Software License
blinker	1.4	Fast, simple object-to-object and broadcast signaling	http://pythonhosted.org/blinker/	MIT License
Brlapi	0.7.0	BrlAPI Client Bindings for Python	http://brltty.app/	LGPL 2.1
cachetools	4.1.0	Extensible memoizing collections and decorators	https://github.com/tkem/cachetools/	MIT
certifi	2019.11.28	Python package for providing Mozilla's CA Bundle.	https://certifi.io/	MPL-2.0
cffi	1.14.0	Foreign Function Interface for Python calling C code.	http://cffi.readthedocs.org	MIT
chardet	3.0.4	Universal encoding detector for Python 2 and 3	https://github.com/chardet/chardet	LGPL
click	7.1.1	Composable command line interface toolkit	https://palletsprojects.com/p/click/	BSD-3-Clause
colorama	0.4.3	Cross-platform colored terminal text.	https://github.com/tartley/colorama	BSD
colored	1.4.2	Simple library for color and formatting to terminal	https://gitlab.com/dslackw/colored	UNKNOWN
colorlog	4.1.0	Log formatting with colors!	https://github.com/borntyping/python-colorlog	MIT License
command-not-found	0.3	UNKNOWN		UNKNOWN
coverage	5.1	Code coverage measurement for Python	https://github.com/nedbat/coveragepy	Apache 2.0
cryptography	2.8	cryptography is a package which provides cryptographic recipes and primitives to Python developers.	https://github.com/pyca/cryptography	BSD or Apache License, Version 2.0
cupshelpers	1.0	Helper functions and classes for using CUPS		UNKNOWN
cycler	0.10.0	Composable style cycles	http://github.com/matplotlib/cycler	BSD
Cython	0.29.20	The Cython compiler for writing C extensions for the Python language.	http://cython.org/	Apache
dbus-python	1.2.16	Python bindings for libdbus	http://www.freedesktop.org/wiki/Software/DBusBindings/#python	Expat (MIT/X11)

1.3 API Reference

You can see below the modules of the *pywebdoc* package.

1.3.1 url

The **url** module manages the URLS. The URLS are ckecked with *requests*.

`url.`**`check_url`**`(`*url*, *log=True*`)`

Return True if no 404 error on url, else False.

`url.`**`open_url`**`(`*url*`)`

Open url if no 404 error is catched.

`class` `url.`**`UrlTemplate`**`(`*template*`)`

Bases: `object`

Manage an url template.

`property template`

Get url template.

`render``(`***kwargs*`)`

Render url template.

`open_url``(`***kwargs*`)`

Render the url template and open it.

1.3.2 packages

The *packages* module manages installed PyPI packages.

class packages.**Package**(*name*)

Bases: object

Manage an installed PyPI package.

property name

Get name.

property url

Get PyPI url.

property version

Get version.

property summary

Get summary.

property home_page

Get home-page.

static get_list()

Return the list of installed packages.

static make_html_list(*filename*)

Make html file with packages list.

1.3.3 pip

The **pip** module can call the `pip` command to get informations about installed PyPI packages.

pip.**call**(**args*)

Call pip command and get result.

pip.**get_installed_packages**()

Return the list of installed packages.

pip.**get_url**(*package*)

Get the home-page url of a PyPI package. The package must be installed.

1.3.4 template

The **template** module manages templates. It uses *jinja2*.

class template.**Templates**(*directory*)

Bases: object

Manage the templates.

property env

Get environment.

get_template(*filename*)

Return a template.

1.3.5 logger

The **logger** module manages the configuration of the logger. It uses *colorlog* to get colored logging messages.

`logger.logger_config(level, name=None)`

Setup the logging environment.

1.3.6 error

The **error** module manages errors of *pywebdoc*.

exception `error.PywebdocError`

Bases: `Exception`

Custom class to raise *pywebdoc* exceptions.

GitHub repository: https://github.com/Quentin18/pywebdoc/

CHAPTER

TWO

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX