ytools3

Release 3.0.1

Dominic Davis-Foster

Jul 12, 2021

CONTENTS

Library for validating `yaml` files against schema and selectively dumping nodes from `yaml` (or `json`) documents in `yaml` or `json` format.

This is a port of `ytools` (https://github.com/yaccob/ytools), which was Python 2 only, to Python 3.

Docs	
Tests	
PyPI	
Activity	
Other	

CHAPTER

ONE

FEATURES

- Output `yaml` as `json` or `python`
- Output `json` as `yaml` or `python` (provided that there are no duplicate mapping entry in the `json` source)
- Extract particular nodes from `yaml` and `json` files.
  - If `yaml` is used as output format (default) the output is a valid `yaml` document.
- Validate `yaml` and `json` documents.
  - The `json-schema` can be provided in `yaml` format as well, which improves readability and writability.
- Preserve order of mapping-keys in `yaml` and `json` output.
- Multi-document support
  - Multiple input files
    - ... as well as multiple `yaml` documents within a file
    - ... and a combination of both

CHAPTER

TWO

INSTALLATION

from PyPI

from GitHub

```
$ python3 -m pip install ytools3 --user
```

```
$ python3 -m pip install git+https://github.com/domdfcoding/ytools3@master --user
```

2.1 API Reference

Library for validating *yaml* files against schema and selectively dumping nodes from *yaml* (or *json*) documents in *yaml* or *json* format.

ytools.**dump**(*datafile*, *path='$'*, *format='yaml'*, *yaml_options='{explicit_start: True, explicit_end: True, allow_unicode: True}'*, *json_options='{indent: 2, encoding: utf-8}'*, *encoding='utf-8'*)

Parameters

- **datafile** (Union[str, Path]) –
- **path** (str) –
- **format** (str) –
- **yaml_options** (str) –
- **json_options** (str) –
- **encoding** (*str*) – Encoding to open the files with.

Return type None

ytools.**validate**(*schemafile*, *datafiles*, *encoding='utf-8'*)

Validate the given datafiles using a schema

Parameters

- **schemafile** (Union[str, Path]) – The json or yaml formatted schema to validate with
- **datafiles** (Iterable[Union[str, Path]]) – An iterable of json or yaml files to validate
- **encoding** (*str*) – Encoding to open the files with.

Return type None

2.2 Downloading source code

The `ytools3` source code resides on publicly accessible GitHub servers, and can be accessed from the following URL: https://github.com/domdfcoding/ytools3"

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

```
$ git clone https://github.com/domdfcoding/ytools3"
> Cloning into 'ytools3'...
> 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

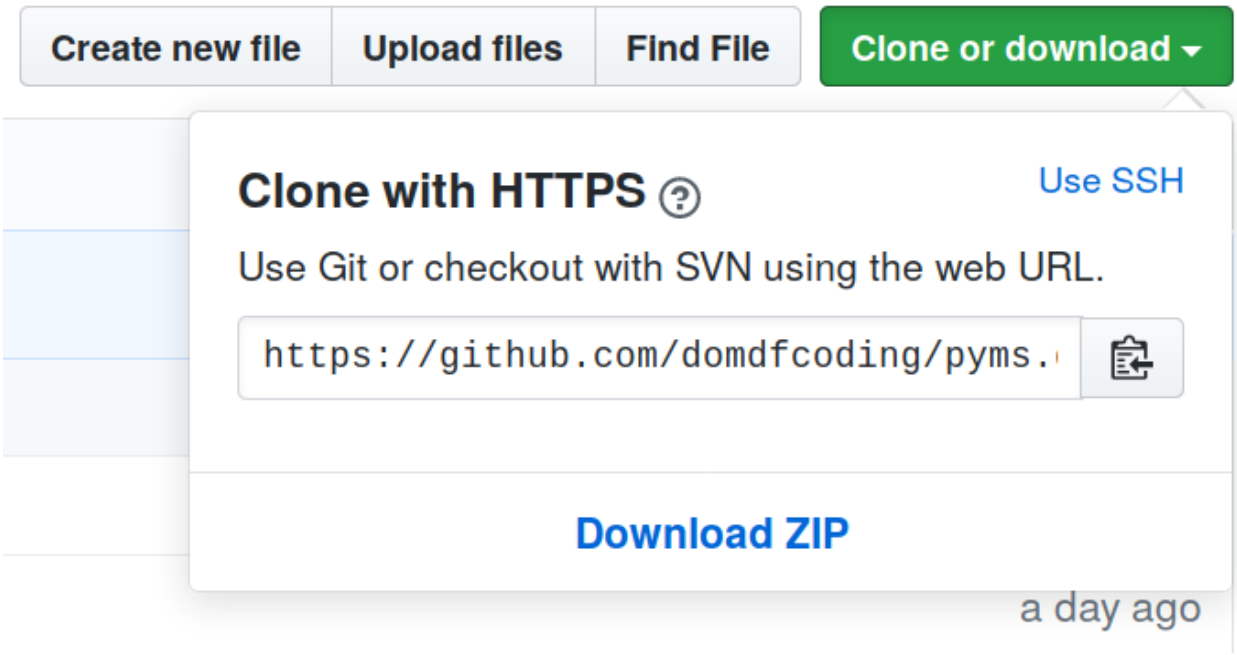

Fig. 1: Downloading a 'zip' file of the source code

2.3 Building from source

To build the `ytools3` package from source using `setuptools`, run the following command:

```
$ python3 setup.py sdist bdist_wheel
```

`setuptools` is configured using the file `setup.py`.

Different formats are available for built distributions

Format	Description	Notes
`gztar`	gzipped tar file (`.tar.gz`)	default on Unix
`bztar`	bzipped tar file (`.tar.bz2`)	
`xztar`	bzipped tar file (`.tar.bz2`)	
`tar`	tar file (`.tar`)	
`zip`	zip file (`.zip`)	default on Windows
`wininst`	self-extracting ZIP file for Windows	
`msi`	Microsoft Installer	

setup.py

```
#!/usr/bin/env python
# This file is managed by `repo_helper`. Don't edit it directly
"""Setup script"""

# stdlib
import sys

# 3rd party
from setuptools import setup

sys.path.append(".")

# this package
from __pkginfo__ import *  # pylint: disable=wildcard-import



setup(
                extras_require=extras_require,
                install_requires=install_requires,
                py_modules=[],
                version=__version__,

                )
```

__pkginfo__.py

```
#  This file is managed by `repo_helper`. Don't edit it directly
#  Copyright (C) 2020 Dominic Davis-Foster <dominic@davis-foster.co.uk>
#
```

(continues on next page)

(continued from previous page)

```
#  This file is distributed under the same license terms as the program it came with.
#  There will probably be a file called LICEN[S/C]E in the same directory as this file.
#
#  In any case, this program is distributed in the hope that it will be useful,
#  but WITHOUT ANY WARRANTY; without even the implied warranty of
#  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
#
# This script based on https://github.com/rocky/python-uncompyle6/blob/master/__pkginfo__.py
#

# stdlib
import pathlib

__all__ = [
                "__copyright__",
                "__version__",
                "modname",
                "pypi_name",
                "__license__",
                "__author__",
                "short_desc",
                "author",
                "author_email",
                "github_username",
                "web",
                "github_url",
                "repo_root",
                "install_requires",
                "extras_require",
                "project_urls",

                "import_name",
                ]

__copyright__ = """
2020 Dominic Davis-Foster <dominic@davis-foster.co.uk>
"""

__version__ = "3.0.1"
modname = "ytools3"
pypi_name = "ytools3"
import_name = "ytools"
__license__ = "Apache2.0"
short_desc = "Library for validating `yaml` files against schema and selectively dumping nodes from `yaml` (or `json`) documents in `yaml` or `json` format."
__author__ = author = "Dominic Davis-Foster"
author_email = "dominic@davis-foster.co.uk"
github_username = "domdfcoding"
web = github_url = "https://github.com/domdfcoding/ytools3"
repo_root = pathlib.Path(__file__).parent
install_requires = (repo_root / "requirements.txt").read_text(encoding="utf-8").split('\n')
extras_require = {'all': []}


```

(continues on next page)

(continued from previous page)

```

project_urls = {
                "Documentation": "https://ytools3.readthedocs.io",
                "Issue Tracker": f"{github_url}/issues",
                "Source Code": github_url,
                }
```

View the Function Index or browse the Source Code.

Browse the GitHub Repository

PYTHON MODULE INDEX

y

INDEX