Docx Utils

Release 0.1.3

Jul 15, 2020

Contents

CHAPTER 1

Overview

docs	
tests	
package	

Creation and manipulation of Open XML documents (mainly docx).

- Free software: MIT license

1.1 Features

This library allow you to:

- Convert Open XML documents into flat OPC format.

1.2 Installation

```
pip install docx-utils
```

1.3 Using the library

Using the library to convert an Open XML document into flat OPC format:

```
>>> from docx_utils.flatten import opc_to_flat_opc
>>> opc_to_flat_opc("sample.docx", "sample.xml")
```

1.4 Command Line Interface (CLI)

Printing the online help:

```
$ docx_utils --help
Usage: docx_utils [OPTIONS] COMMAND [ARGS]...

  Docx utilities

Options:
  --version  Show the version and exit.
  --help     Show this message and exit.

Commands:
  flatten  Convert an Open XML document into flat OPC format.
```

Converting an Open XML document into flat OPC format:

```
$ docx_utils flatten sample.docx sample.xml
Converting 'sample.docx' to flat XML...
Conversion done: 'sample.xml'.
```

1.5 Documentation

https://docx-utils.readthedocs.io/en/latest/

1.6 Development

To run the all tests run:

```
tox
```

Note, to combine the coverage data from all the tox environments run:

Windows	`set PYTEST_ADDOPTS=--cov-append` `tox`
Other	`PYTEST_ADDOPTS=--cov-append tox`

CHAPTER 2

Installation

At the command line:

```
pip install docx-utils
```

To use this library in your application, add the dependency in the `setup.py`:

```
setup(
    name="my_app",
    version="1.0.3",
    install_requires=[
        'docx-utils',
        ...
    ],
    ...
)
```

Don't forget to update your virtualenv:

```
pip install -e .
```

The `docx_utils` library should be available, check it with:

```
docx_utils --version
```

CHAPTER 3

API

This part of the documentation covers all the interfaces of the Docx Utils Library.

3.1 Docx-utils Library

This library allow you to:

- Convert Open XML document to **flat** OPC format.

3.2 Exceptions

Exception hierarchy for the docx-utils package.

exception docx_utils.exceptions.**DocxUtilsException**
Base exception of the docx-utils package.

exception docx_utils.exceptions.**UnknownContentTypeError**(*opc_path*, *uri*)
Exception raised during Microsoft Office document parsing when a part can't be resolved.

fmt = "Cannot parse the Microsoft Office document '{opc_path}': the content-type of t

opc_path

uri

3.3 Command line interface (CLI)

3.3.1 Overview

This module defines the main command line interface (CLI).

3.4 Docx to flat XML converter

This converter is inspired from Eric White's article: Transforming Open XML Documents to Flat OPC Format.

> This post describes the process of conversion of an Open XML (OPC) document into a Flat OPC document, and presents the C# function, OpcToFlat.

The function *opc_to_flat_opc()* is used to convert an Open XML document (.docx, .xlsx, .pptx) into a flat OPC format (.xml).

class docx_utils.flatten.**ContentTypes**
ContentTypes contained in a "[Content_Types].xml" file.

NS = {'ct': u'http://schemas.openxmlformats.org/package/2006/content-types'}

parse_xml_data(*data*)

resolve(*part_name*)

class docx_utils.flatten.**PackagePart**(*uri*, *content_type*, *data*)

content_type
Alias for field number 1

data
Alias for field number 2

uri
Alias for field number 0

docx_utils.flatten.**iter_package**(*opc_path*, *on_error='ignore'*)
Iterate a Open XML document and yield the package parts.

Parameters

- **opc_path** (*str*) – Microsoft Office document to read (.docx, .xlsx, .pptx)
- **on_error** (*str*) – control the way errors are handled when a part URI cannot be resolved:
  - 'ignore": ignore the part,
  - 'strict': raise an exception.

Returns Iterator which yield package parts

Raises ***UnknownContentTypeError*** – if a part URI cannot be resolved.

docx_utils.flatten.**opc_to_flat_opc**(*src_path*, *dst_path*, *on_error='ignore'*)
Convert an Open XML document into a flat OPC format.

Parameters

- **src_path** (*str*) – Microsoft Office document to convert (.docx, .xlsx, .pptx)
- **dst_path** (*str*) – Microsoft Office document converted into flat OPC format (.xml)
- **on_error** (*str*) – control the way errors are handled when a part URI cannot be resolved:
  - 'ignore": ignore the part,
  - 'strict': raise an exception.

CHAPTER 4

Contributing

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

4.1 Bug reports

When 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.

4.2 Documentation improvements

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

4.3 Feature requests and feedback

The best way to send feedback is to file an issue at https://github.com/tantale/docx_utils/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 code contributions are welcome :)

4.4 Development

To set up *docx_utils* for local development:

1. Fork docx_utils (look for the "Fork" button).
2. Clone your fork locally:

```
git clone git@github.com:your_name_here/docx_utils.git
```

3. Create a branch for local development:

```
git checkout -b feature/name-of-your-feature  # or
git checkout -b fix/name-of-your-bugfix
```

   Now you can make your changes locally.

4. When you're done making changes, run all the checks, doc builder and spell checker with tox one command:

```
tox
```

5. Commit your changes and push your branch to GitHub:

```
git add .
git commit -m "Your detailed description of your changes."
git push origin feature/name-of-your-feature  # or
git push origin fix/name-of-your-bugfix
```

6. Submit a pull request through the GitHub website.

4.4.1 Pull Request Guidelines

If you need some code review or feedback while you're developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run `tox`)[1].
2. Update documentation when there's new API, functionality etc.
3. Add a note to `CHANGELOG.rst` about the changes.
4. Add yourself to `AUTHORS.rst`.

4.4.2 Tips

To run a subset of tests:

```
tox -e envname -- pytest -k test_myfeature
```

To run all the test environments in *parallel* (you need to `pip install detox`):

```
detox
```

[1] If you don't have all the necessary python versions available locally you can rely on Travis - it will run the tests for each change you add in the pull request.
It will be slower though...

CHAPTER 5

Authors

- Laurent LAPORTE - https://github.com/tantale

CHAPTER 6

Changelog

All notable changes to this project will be documented in this file.

The format is based on Keep a Changelog and this project adheres to Semantic Versioning.

6.1 v0.1.3 (2020-07-15)

6.1.1 Fixed

- Correct the project's dependencies: Enum34 is only required for Python versions < 3.4.
- Add the `exceptions` module: Exception hierarchy for the docx-utils package.
- Fix #1:
  - Add the *on_error* option in the `opc_to_flat_opc()` function in order to ignore (or raise an exception) when a part URI cannot be resolved during the Microsoft Office document parsing.
  - Change the command line interface: add the `--on-error` option to handle parsing error.

6.1.2 Other

- Continuous Integration: add configurations for Python 3.7 and Python 3.8.

6.2 v0.1.2 (2018-07-26)

6.2.1 Fixed

- Drop support for PyPy: it seams that lxml is not available for this Python implementation.

- Drop support for Python 3.7: this Python version is not yet available on all platform. However, it is known to work on Ubuntu with the python-3.7-dev release.

6.2.2 Other

- Use the pseudo-tags `start-exclude`/`end-exclude` in `CHANGELOG.rst` and `README.rst` to exclude text from the generated `PKG-INFO` during setup.

6.3 v0.1.1 (2018-07-25)

6.3.1 Fixed

- Fix wheel version on PyPi.

6.4 v0.1.1 (2018-07-24)

6.4.1 Added

- First release.

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Python Module Index

d

Index