docx2csv documentation

Release 0.1.0

Ivan Begtin

Jan 14, 2022

Contents

1 Examples **3**

2 Code **5**

3 Requirements **7**

4 Acknowledgements **9**

5 Command line **11**

5.1 Code 11
5.2 Requirements 12
5.3 Acknowledgements 12
5.4 Documentation 12
5.5 Indices and tables 14

Usage: docx2csv [OPTIONS] FILENAME

docx to csv convertor (http://github.com/ivbeg/docx2csv) Extracts tables from DOCX files as CSV or XLSX.

Use command: "docx2csv convert <filename>" to run extraction. It will create files like filename_1.csv, filename_2.csv for each table found.

Options:

--format TEXT	Output format: CSV, XLSX
--singlefile TEXT	Outputs single XLS file with multiple sheets: True or False
--sizefilter INTEGER	Filters table by size number of rows
--help	Show this message and exit.

CHAPTER 1

Examples

docx2csv –format csv –sizefilter 3 CP_CONTRACT_160166.docx

Extracts tables from file CP_CONTRACT_160166.docx with number of rows > 3 and saves results as CSV files.

CHAPTER 2

Code

Function “extract_tables” returns list of tables from docx file and function “extract” extracts tables as xlsx, xls or csv file. If ‘csv

```
>>> from docx2csv import extract_tables, extract
>>> tables = extract_tables('some_file.docx')
```

returns list of tables >>> extract(filename=’some_file.docx’, format=”xlsx”, output=’some_file.xlsx’) saves all tables from some_file.docx to some_file.xlsx

```
>>> extract(filename='some_file.docx', format="csv", singlefile=False)
saves all tables from some_file.docx to some_file_1.csv, some_file_2.csv and etc.
```

CHAPTER 3

Requirements

- click https://github.com/pallets/click
- xlwt https://github.com/python-excel/xlwt
- python-docx https://github.com/python-openxml/python-docx
- openpyxl https://bitbucket.org/openpyxl/openpyxl/src

CHAPTER 4

Acknowledgements

Thanks to Vsevolod Oparin (https://www.facebook.com/vsevolod.oparin) for optimized “extract_table” code

CHAPTER 5

Command line

‘‘‘ Usage: docx2csv [OPTIONS] FILENAME

docx to csv convertor (http://github.com/ivbeg/docx2csv) Extracts tables from DOCX files as CSV or XLSX.

Use command: “docx2csv convert <filename>” to run extraction. It will create files like filename_1.csv, filename_2.csv for each table found.

Options:

--format TEXT	Output format: CSV, XLSX
--singlefile TEXT	Outputs single XLS file with multiple sheets: True or False
--sizefilter INTEGER	Filters table by size number of rows
--help	Show this message and exit.

‘‘‘ ## Examples

docx2csv –format csv –sizefilter 3 CP_CONTRACT_160166.docx

Extracts tables from file CP_CONTRACT_160166.docx with number of rows > 3 and saves results as CSV files.

5.1 Code

5.1.1 Popular Formats

Function ‘parse’ mimics default behavior of dateparser ‘parse’ function. Except that it is part of DateParser class, not standalone function.

```
>>> from docx2csv import extract_tables, extract
>>> tables = extract_tables('some_file.docx')
```

returns list of tables >>> extract(filename=’some_file.docx’, format=”xlsx”, output=’some_file.xlsx’) saves all tables from some_file.docx to some_file.xlsx

5.2 Requirements

- click https://github.com/pallets/click
- xlwt https://github.com/python-excel/xlwt
- python-docx https://github.com/python-openxml/python-docx
- openpyxl https://bitbucket.org/openpyxl/openpyxl/src

5.3 Acknowledgements

Thanks to Vsevolod Oparin (https://www.facebook.com/vsevolod.oparin) for optimized "extract_table" code

5.4 Documentation

Contents:

5.4.1 Installation

At the command line:

```
$ pip install docx2csv
```

Or, if you don't have pip installed:

```
$ easy_install docx2csv
```

If you want to install from the latest sources, you can do:

```
$ git clone https://github.com/ivbeg/docx2csv.git
$ cd docx2csv
$ python setup.py docx2csv
```

5.4.2 Contributing

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

You can contribute in many ways:

Types of Contributions

Report Bugs

Report bugs at https://github.com/ivbeg/docx2csv/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.

Fix Bugs

Look thrgough the GitHub issues for bugs. Anything taged with "bug" is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with "feature" is open to whoever wants to implement it. We encourage you to add new languages to existing stack.

Write Documentation

docx2csv could always use more documentation, whether as part of the official docx2csv docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/ivbeg/docx2csv/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 contributions are welcome :)

Get Started!

Ready to contribute? Here's how to set up *docx2csv* for local development.

1. Fork the *docx2csv* repo on GitHub.
2. Clone your fork locally:
3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv docx2csv
$ cd docx2csv/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ pip install -r tests/requirements.txt # install test dependencies
$ flake8 docx2csv tests
$ nosetests
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv. (Note that we use `max-line-length = 100` for flake8, this is configured in `setup.cfg` file.)

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

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

7. Submit a pull request through the GitHub website.

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/ivbeg/docx2csv/pull_requests and make sure that the tests pass for all supported Python versions.
4. Follow the core developers' advice which aim to ensure code's consistency regardless of variety of approaches used by many contributors.
5. In case you are unable to continue working on a PR, please leave a short comment to notify us. We will be pleased to make any changes required to get it done.

5.4.3 Credits

Committers

- Ivan Begtin
- Vsevolod Oparin

5.4.4 History

0.1.0 (2018-01-14)

- First public release on PyPI and updated github code

5.5 Indices and tables

- genindex
- modindex

- search