
NotebookToAll Documentation

Release 0.2.3

Jeff Hale

Feb 08, 2022

CONTENTS:

1	NotebookToAll	1
1.1	Quick Start	1
2	Installation	3
2.1	Stable release	3
2.2	From sources	3
3	Usage	5
4	notebooktoall	7
4.1	notebooktoall package	7
5	Contributing	9
5.1	Types of Contributions	9
5.2	Get Started!	10
5.3	Pull Request Guidelines	11
5.4	Tips	11
5.5	Deploying	11
6	Credits	13
6.1	Development Lead	13
6.2	Contributors	13
7	History	15
7.1	0.0.1 (2019-02-11)	15
8	Indices and tables	17
	Python Module Index	19
	Index	21

NOTEBOOKTOALL

The coolest way to turn a Jupyter Notebook into sharable files.

Turn your Jupyter Notebook into a .py executable file and an .html file with a one-line command. You can create both or either file type at the same time! Under the hood, this package uses Jupyter nbconvert.

nbconvert is great for command line transformations, but it takes a bit of work to use in a program. NotebookToAll to the rescue.

1.1 Quick Start

1.1.1 Install

Install from PyPI.

```
pip install notebooktoall
```

1.1.2 Use

Make sure your notebook doesn't have magic commands in it if you want to create an executable .py script.:

```
from notebooktoall.transform import transform_notebook

transform_notebook(ipynb_file="my_jupyter_notebook.ipynb", export_list=["html", "py"])
```

Run your code and your .html and .py files should appear in your current working directory.

You can pass a Jupyter notebook URL to transform_notebook().

1.1.3 License

- Free software: GNU General Public License v3

1.1.4 Docs

- Documentation: <https://notebooktoall.readthedocs.io>.

1.1.5 Credits:

The [Cookiecutter audreyr/cookiecutter-pypackage](#) project template.

INSTALLATION

2.1 Stable release

To install NotebookToAll, run this command in your terminal:

```
$ pip install notebooktoall
```

This is the preferred method to install notebooktoall, as it will always install the most recent stable release.

If you don't have `pip` installed, this [Python installation guide](#) can guide you through the process.

2.2 From sources

The sources for notebooktoall can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/notebooktoall/notebooktoall
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/notebooktoall/notebooktoall/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```


USAGE

To use NotebookToAll in a project:

Make sure your notebook doesn't have magic commands in it if you want to create an executable .py script.

```
from notebooktoall.transform import transform_notebook

transform_notebook(ipynb_file="my_jupyter_notebook.ipynb", export_list=["html
↪", "py"])
```

Run your program and your .html and .py files should appear in your current working directory.

You can also pass a Jupyter notebook URL to transform_notebooks().

NOTEBOOKTOALL

4.1 notebooktoall package

4.1.1 Submodules

4.1.2 notebooktoall.transform module

Transform Jupyter Notebook to executable python script and/or html file.

`notebooktoall.transform.get_notebook(ipynb_file)`

Convert Jupyter notebook from url or directory to a NotebookNode object.

Parameters `ipynb_file` (*str*) – url or path of notebook

Returns NotebookNode object

`notebooktoall.transform.transform_notebook(ipynb_file, export_list=['html', 'py'])`

Transform notebook to specified document types.

Parameters

- `ipynb_file` (*str*) – url or path of Jupyter notebook to be converted
- `export_list` (*list of str*s) – output file types. Defaults: html, py.

Returns None

`notebooktoall.transform.write_files(export_list, nb_node, file_name)`

Export and write files from a notebook node.

Parameters

- `export_list` (*list of strings*) –
- `nb_node` (*nbformat node object*) –
- `file_name` (*str*) –

Returns None

4.1.3 Module contents

Top-level package for notebooktoall.

CONTRIBUTING

Contributions are welcome and greatly appreciated!

You can contribute in many ways:

Please ensure you follow our Code of Conduct.

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at <https://github.com/notebooktoall/notebooktoall/issues>.

If you are reporting a bug, please include:

- Your operating system name and version, if applicable.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” and “help wanted” is open to whoever wants to implement it.

5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with “enhancement” and “help wanted” is open to whoever wants to implement it.

5.1.4 Write Documentation

notebooktoall could always use more documentation, whether as part of the official notebooktoall docs, in docstrings, or even on the web in blog posts. Submit Feedback ~~~~~

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

5.2 Get Started!

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

1. Fork the *notebooktoall* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/notebooktoall.git
```

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 notebooktoall
$ cd notebooktoall/
$ 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:

```
$ flake8 notebooktoall tests
$ python setup.py test or py.test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

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.

5.3 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. The pull request should work for Python 3.6 and for PyPy. Check https://travis-ci.org/notebooktoall/notebooktoall/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ py.test tests.test_notebooktoall
```

5.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

CREDITS

6.1 Development Lead

- Jeff Hale <jeffmshale@gmail.com>

6.2 Contributors

None yet. Why not be the first?

HISTORY

7.1 0.0.1 (2019-02-11)

- First release on PyPI.

7.1.1 0.1.0 (2019-02-11)

- Add Code of Conduct
- Improve docs
- Fix badges
- Cover multiple python versions

7.1.2 0.1.1 (2019-02-11)

- Bump version

7.1.3 0.1.2 (2019-02-11)

- Add missing dependency
- Update ReadTheDocs usage

7.1.4 0.1.3 (2019-02-11)

- Improve rst doc formatting
- Wheel

7.1.5 0.1.4 (2019-02-11)

- Nada

7.1.6 0.1.5 (2019-02-11)

- Add requirements to setup.py

7.1.7 0.1.6 (2019-02-12)

- One requirements file

7.1.8 0.1.7 (2019-02-12)

- Fix setup.py

7.1.9 0.1.8 (2019-02-13)

- Update coverage

7.1.10 0.2.0 (2019-02-14)

- API change to transform_notebook() from transform_notebooks()
- More Tests

7.1.11 0.2.1 (2019-02-15)

- RTD improvements
- Fewer packages required for install
- Doc fixes

7.1.12 0.2.2 (2019-02-18)

- Simplify structure

7.1.13 0.2.3 (2019-02-18)

- Fix docs and package

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

n

`notebooktoall`, [8](#)

`notebooktoall.transform`, [7](#)

INDEX

G

`get_notebook()` (*in module notebooktoall.transform*), 7

M

module
 notebooktoall, 8
 notebooktoall.transform, 7

N

notebooktoall
 module, 8
notebooktoall.transform
 module, 7

T

`transform_notebook()` (*in module notebooktoall.transform*), 7

W

`write_files()` (*in module notebooktoall.transform*), 7