Jupyter Notebook Tools for Sphinx

Release 0.1.0

Matthias Geier

November 29, 2015

Contents

For now, this is just a test to see if `*.ipynb` files can be used as Sphinx sources using `source_parsers`.

Documentation: http://nbsphinx.rtfd.org/

Code: http://github.com/mgeier/nbsphinx/

1 Links

https://github.com/ngoldbaum/RunNotebook

https://bitbucket.org/yt_analysis/yt-doc/src/default/extensions/notebook_sphinxext.py

https://github.com/matthew-brett/perrin-academy/blob/master/sphinxext/notebook_sphinxext.py

http://sphinx-ipynb.readthedocs.org/

http://dongweiming.github.io/divingintoipynb_nikola/posts/nbconvert.html

https://github.com/ipython/ipython/issues/4936

https://mail.scipy.org/pipermail/ipython-user/2013-December/013490.html

https://github.com/ipython/nbconvert/pull/35

https://github.com/matthew-brett/brole

https://github.com/perrette/dimarray/blob/master/docs/scripts/nbconvert_to_rst.py

https://github.com/matthew-brett/nb2plots

https://github.com/getpelican/pelican-plugins/blob/master/liquid_tags/notebook.py

https://github.com/jupyter/nbconvert/issues/47

http://hplgit.github.io/doconce/doc/web/index.html

http://sphinx-doc.org/extdev/

https://github.com/sphinx-doc/sphinx/issues/1907

2 Example Pages

2.1 An Example Notebook

This notebook is meant for testing conversion to other formats.

It contains Markdown cells and code cells with different kinds of outputs.

Markdown

We can write *in italics* (same *with underscores*), **in boldface** (same **with undescores**) and [STRIKEOUT:strikethrough]. We also can write `preformatted text`.

Equations

Equations can be formatted really nicely, either inline, like $e^{i\pi} = -1$, or on a separate line, like

$$\int_{-\infty}^{\infty} f(x)\delta(x - x_0)dx = f(x_0)$$

Code

We can also write code with nice syntax highlighting:

```
print("Hello, world!")
```

Tables

A	B	A and B
False	False	False
True	False	False
False	True	False
True	True	True

Images

Jupyter notebook icon (local):

Python logo (local):

Jupyter logo (remote):

Python logo (remote):

Code Cells

An empty code cell:

A cell with no output:

```
None
```

A simple output:

```
6 * 7
```

```
42
```

The standard output stream:

```
print('Hello, world!')
```

```
Hello, world!
```

Normal output + standard output

```
print('Hello, world!')
6 * 7
```

```
Hello, world!
```

```
42
```

The standard error stream is highlighted and displayed just below the code cell. The standard output stream comes afterwards (with no special highlighting). Finally, the “normal” output is displayed.

```
import logging
logging.warning('I am a warning and I will appear on the standard error stream')
print('I will appear on the standard output stream')
'I am the "normal" output'
```

```
WARNING:root:I am a warning and I will appear on the standard error stream
```

```
I will appear on the standard output stream
```

```
'I am the "normal" output'
```

Special Display Formats

See IPython example notebook.

TODO: tables? e.g. Pandas DataFrame?

```
from IPython.display import display, Image, SVG, Math, YouTubeVideo
```

Local Image Files

```
i = Image(filename='images/notebook_icon.png')
i
```


```
display(i)
```

For some reason this doesn't work with `Image(...)`:

```
SVG(filename='images/python_logo.svg')
```

Image URLs

```
Image(url='https://www.python.org/static/img/python-logo-large.png')
```

```
Image(url='https://www.python.org/static/img/python-logo-large.png', embed=True)
```

```
Image(url='http://jupyter.org/assets/nav_logo.svg')
```

```
Image(url='https://www.python.org/static/favicon.ico')
```

```
Image(url='http://python.org/images/python-logo.gif')
```

Math

```
eq = Math(r"\int_{-\infty}^\infty f(x) \delta(x - x_0) dx = f(x_0)")
eq
```

$$\int_{-\infty}^{\infty} f(x)\delta(x - x_0)dx = f(x_0)$$

```
display(eq)
```

$$\int_{-\infty}^{\infty} f(x)\delta(x - x_0)dx = f(x_0)$$

```
%%latex
\begin{equation}
\int_{-\infty}^\infty f(x) \delta(x - x_0) dx = f(x_0)
\end{equation}
```

$$\int_{-\infty}^{\infty} f(x)\delta(x - x_0)dx = f(x_0)(1)$$

```
YouTubeVideo('iV2ViNJFZC8')
```

Raw Cells

Raw cells are interpreted as reStructuredText and parsed by Sphinx.

Back to main page

2.2 A Pre-Executed Notebook

Notebooks with no outputs are automatically executed during the Sphinx build process. If, however, there is at least one output cell present, the notebook is not evaluated and included as is.

This can be useful for the following use cases.

Long-Running Cells

If you are doing some very time-consuming computations, it might not be feasible to re-execute the notebook every time you build your Sphinx documentation.

So just do it once - when you happen have the time - and then just keep the output.

```
import time
```

```
%time time.sleep(60 * 60)
6 * 7
```

```
CPU times: user 160 ms, sys: 56 ms, total: 216 ms
Wall time: 1h 1s
```

```
42
```

Rare Libraries

You might have created results with a library that's hard to install and therefore you have only managed to install it on one very old computer in the basement, so you probably cannot run this whenever you build your Sphinx docs.

```
from a_very_rare_library import calculate_the_answer
```

```
calculate_the_answer()
```

```
42
```

Exceptions

If an exception is raised during the Sphinx build process, it is stopped. If you want to show to your audience how an exception looks like, you'll have to run the notebook beforehand and include at least one output.

```
1 / 0
```

```
---------------------------------------------------------------------------

ZeroDivisionError                         Traceback (most recent call last)

<ipython-input-5-b710d87c980c> in <module>()
```

```
----> 1 1 / 0


ZeroDivisionError: division by zero
```

2.3 A Notebook in a Sub-Directory

Let's see if links to local images work:

```
from IPython.display import Image
Image(filename='../images/notebook_icon.png')
```

2.4 A Normal reStructuredText File

This is a normal RST file.

Note: Those still work!

There is also `orphan`, just for the sake of it.