Index

September 18, 2016

1 Jupyter Site

JupyterSite allows you to easily build a suite of rendered documents from Jupyter notebooks. Jupyter notebooks go in, html/markdown pages, PDFs, slideshows, and tex comes out. If you use this in a Github repository, it will even build a Github pages site. This also shows how to make the notebooks easily accessible in Julia.

It just takes one button, simplifying the entire process. The `publish.sh` file does all the heavy lifting, showing you how to do all of this (any of the conversions can be disabled by commenting out the appropriate lines).

1.1 Example Site

An example of the product this package generates can be found at: https://github.com/UCIDataScienceInitiative/IntroToJulia .

The site that this generates is: http://UCIDataScienceInitiative.github.io/IntroToJulia/

1.2 Installation

To use this, first install Jupyter (note: Python must be installed)

```
pip install jupyter
```

For the PDF output to work, a distribution of LaTeX must be installed. Also, we will need pandoc:

```
pip install pandoc
```

1.3 Using the Package

1.3.1 Downloading the Template

The easiest way to get started is to clone the package repository using Git:

```
git clone https://github.com/ChrisRackauckas/JupyterSite
```

1.3.2 The Notebooks

The files which the site is built from are the notebooks that are included in the Notebooks folder. The `Index.ipynb` file build the site index (the first page shown in the webpage) and additional `.ipynb` files are rendered in the various forms. Thus to add your own content, simply add the `.ipynb` files to this directory.

Note that notebooks have to be setup for slides in order for the slideshow to work. Inside the notebook, use view > CellToolbar > Slideshow and set the appropriate blocks to slides.

1.3.3 Building The Site

To build the site files, use the `publish.sh` file. To do so, go to the top directory of the repository and use the command:

```
sh publish.sh
```

If your Git is correctly setup, this will render the files and upload the files to Github. Upon success, your `index.html` will be available at:

```
http://<github-username>.github.io/<repository-name>/
```

For example, my user name is ChrisRackauckas and this repository is named JupyterSite. Therefore the site for that this makes is at

```
http://ChrisRackauckas.github.io/JupyterSite/
```

For organizations, the user name is replaced with the organization name.

1.4 Easy Usage from Julia

This repository is also setup as a Julia package repository. To have users easily open up the Jupyter notebooks, they can use the commands from within Julia:

```
In [ ]: Pkg.add("IJulia") # use once to install IJulia
        Pkg.clone("https://github.com/ChrisRackauckas/JupyterSite") # Change this t
        using IJulia
        notebook(dir=Pkg.dir("JupyterSite")*"/Notebooks")
```

This will open up the Jupyter notebook at the location of your notebooks

1.4.1 Additional Feature: Continuous Integration Testing

One can use Julia's continuous integration testing framework to test your notebooks. To do so, check out this blog post for how to setup Julia CI and NBinclude.jl for how to include notebook files into Julia. Use this to add tests to the test/runtests.jl file. To run the tests locally, use the commands

```
In [ ]: Pkg.test("JupyterSite")
```

2 Example Output

The example notebooks can be found at:
https://github.com/ChrisRackauckas/JupyterSite/blob/master/Notebooks

2.1 HTML

https://chrisrackauckas.github.io/JupyterSite/
https://chrisrackauckas.github.io/JupyterSite/Html/GithubIntroduction.html

2.2 Slides

https://chrisrackauckas.github.io/JupyterSite/
https://chrisrackauckas.github.io/JupyterSite/Slides/GithubIntroduction.html

2.3 Markdown

https://github.com/ChrisRackauckas/JupyterSite/tree/master/Markdown

2.4 LaTeX

https://github.com/ChrisRackauckas/JupyterSite/tree/master/Tex

2.5 PDF

https://github.com/ChrisRackauckas/JupyterSite/tree/master/Pdfs