
Jupinx Documentation

Release 0.0.1

QuantEcon Development Team

May 10, 2020

Contents:

1	jupinx-quickstart	3
2	Jupinx <i>cmd</i> line utility	5
3	Custom Configuration through sphinxcontrib-jupyter	9
4	Credits	11
5	LICENSE	13
6	Indices and tables	15

Jupinx is a collection of utilities and tools for Jupyter and Sphinx

A quickstart utility has been developed to help users get setup quickly with Sphinx, configured in a way to get building collections of Jupyter notebooks quickly.

- *Installation*
- *Running `jupinx-quickstart`*
- *Directory structure*

1.1 Installation

To install `jupinx`:

```
pip install jupinx
```

or you can upgrade to the latest version using:

```
pip install --upgrade jupinx
```

Note: Windows is currently not tested or supported. See [Issue #7](#)

1.2 Running `jupinx-quickstart`

Once, `jupinx` is installed, to run the `jupinx` quickstart program you can run:

```
jupinx-quickstart
```

on a terminal.

The `jupinx-quickstart` will:

1. setup a `directory structure` for your project
2. check for `sphinxcontrib-jupyter` and `sphinxcontrib-bibtex` installation
3. construct `Makefile` and `conf.py` files
4. construct a parent document `source/index.rst`
5. setup the project to use the `minimal` theme

after running the quickstart you may run:

1. `make jupyter` to build the project as notebooks
2. `make website` to build the project as a website (via `sphinxcontrib-jupyter`)
3. `make pdf` to build the project as a pdf (via `sphinxcontrib-jupyter`)

Note: The `quickstart` sets up the *Makefile* with some *conf.py* setting overrides to enable building *jupyter* and *website* (rather than via a specific builder)

1.3 Directory structure

The following directory structure is adopted during the setup:

- `./`
 - `source`: where source RST files should be added
 - `source/_static`: where `_static` assets such as figures and images are kept
 - `theme`: allows you to customise builders using themes and templates
 - `Makefile`: provides `make` commands for compiling the project
 - `conf.py`: provides configuration for `sphinx-build`

Note: `sphinx` is quite flexible in setting up a project in a way that suits your workflow. If you want to change directory structure this is likely possible but you will need to update your *Makefile* after the quickstart is finished. Please refer to [sphinx docs](#) for further information.

Jupinx *cmd* line utility

- *Installation*
- *Usage*
- *Options*

The *jupinx* command line utility.

Note: this utility currently takes a zero-configuration approach. If you need to modify the behaviour of *sphinxcontrib-jupyter* then you need to update *conf.py* file in your sphinx project.

2.1 Installation

To install *jupinx*:

```
pip install jupinx
```

to upgrade your current installation to the latest version:

```
pip install jupinx --upgrade
```

2.2 Usage

To build a collection of notebooks using *jupinx*:

```
jupinx --notebooks <PATH-PROJECT-DIRECTORY>
```

or

```
jupinx -n <PATH-PROJECT-DIRECTORY>
```

Note: Many users will run *jupinx* at the root level of a repository. this can be done by specifying `jupinx --notebooks`. The directory specification is optional in this case.

It is also possible to build a full website. This option makes use of Jupyter Notebooks ability to execute code so output is not required in any of the source files. The website can be completely built (including all code and generated components).

```
jupinx --website <PATH-PROJECT-DIRECTORY>
```

Note: There is currently **no** default template provided for constructing websites. This needs to be provided in the future to allow building websites out of the box with a default theme.

or

```
jupinx -w <PATH-PROJECT-DIRECTORY>
```

documentation regarding options for building websites can be found [here](#)

All command line options available can be listed using the help flag:

```
jupinx --help
```

or

```
jupinx -h
```

2.3 Options

The typical usage for *jupinx* is:

```
jupinx [OPTIONS] <DIRECTORY> [ADDITIONAL OPTIONS]
```

The following **options** are provided:

-h, --help	show this help message and exit
-c, --clean	clean build directory
-j, --jupyternb	open jupyter to view notebooks
-n, --notebooks	compile RST files to Jupyter notebooks
-d, --pdf	compile RST files to PDF files
-s, --server	open html server to view website
-t, --coverage-tests	compile coverage report for project
-w, --website	compile website
--version	show program's version number and exit

The following **additional options** are provided:

-p [PARALLEL], -parallel [PARALLEL] Specify the number of workers for parallel execution (Default: `-parallel` will result in `-parallel=2`)

-f [FILES [FILES ...]], -files [FILES [FILES ...]] specify files for compilation

Custom Configuration through sphinxcontrib-jupyter

This project depends on [sphinxcontrib-jupyter](#) to enhance `sphinx` to build and work with Jupyter notebooks.

Full documentation for the extension can be found [here](#)

There are many configuration settings that can adjust the compilation behaviour of your project.

3.1 An Example

Let's say you have a collection of notebooks that you would like pre-executed. You can do this by modifying the `conf.py` file to enable notebook execution.

Add the following in the `conf.py` in the *jupyter* options section:

```
jupyter_execute_notebooks = True
```

as documented [here](#)

and let's imagine some of your documents produce a file required by a future document in your collection. An execution dependency can be added to your project by specifying:

```
jupyter_dependency_lists = {  
    'lecture2' : ['lecture1']  
    'lecture3' : ['lecture1']  
}
```

in the `conf.py` file as documented [here](#)

CHAPTER 4

Credits

This project is supported by [QuantEcon](#)

Many thanks to the lead developers of this project.

- [@AakashGfude](#)

Copyright © 2019 QuantEcon Development Team: BSD-3 All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

CHAPTER 6

Indices and tables

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