# Autogenerate documentation with sphinx

This method is for generating and managing documentation for existing or new projects.

*Note: Cookiecutter templates may have Sphinx bundled into them already.*

1. Create your environment and project folder
2. Install sphinx
3. Create ‘docs’ folder and cd to docs. Then run ‘sphinx-quickstart’. This will create the following files.

* conf.py
* index.rst
* Makefile
* make.bat

conf.py – is your configuration file. The settings you chose during the quickstart can be changed and other setting can be added.

Index.rst – is your index page and is converted to an index.html through the build process.

Makefile and make.bat – contain the functions that are called when you build your html pages.

These following commands are used to 1) clean your ‘build’ directory and 2) create your html files in it. You will run them at the end of the tutorial.

‘make clean’

‘make html’

# Generating the HTML – RST to HTML

If you already have your documentation written in reStructuredText (RST) then you can paste that into the index.rst file and add any .rst files you would like to link to from index.rst to the same directory. If you do this you can skip to the last section of this tutorial where you will run ‘make html’.

If, you have Word Documents, Jupyter Notebooks, or Markdown files that you want to create documentation from, you need to first convert them to RST then generate the html.

## DOCX, Jupyter Notebooks, and MarkDown to RST

While there are a couple of different methods depending on the file type of your documentation, in each case you will need to first convert everything to RST.

### docx to rst

Move Word docx to docs/source/worddocs

Pandoc.exe -f docx filename.docx -t rst -o filename.rst

Unzip filename.docx (need bash to use unzip)

Move filename.rst to docs/source level with index.rst

Move media folder with images in it to docs/source/\_static

Open filename.rst and search for ‘image’.

Fix paths to where you images are located e.g. ‘image:: \_static/media/image2.jpeg’

### ipynb to rst

Install nbsphinx

# Edit your conf.py and add 'nbsphinx' to extensions.

Extensions = [

‘sphinx.ext.autodoc’,

‘sphinx.ext.doctest’,

’sphinx.ext.intersphinx’,

’sphinx.ext.autosummary’,

’sphinx.ext.mathjax’,

’nbsphinx’

]

# Add type of source files

source\_suffix = [‘.rst’, ‘.ipynb’]

Edit your index.rst and add the names of your \*.ipynb files to the toctree.

Make sure to put your .ipynb files in the docs/source folder

When you run make html the jupyter notebook will be converted to html and appear just as a converted Word document.

### MD to rst

Edit conf.py

‘Import recommonmark.Parser’

# Add a sourcefile parser for markdown

Source\_parsers = {

‘.md’: ‘recommonmark.parser.CommonMarkParser’,

}

# Add type of source files

source\_suffix = [‘.rst’, ‘.md’]

# indes.rst – sample with mix and matched sources

.. toctree::

:maxdepts: 2

:caption: Contents:

Sample\_doc.rst

Markdown\_doc.md

Notebook.ipynb

# Conf.py – sample

Extensions = [

‘sphinx.ext.autodoc’,

‘sphinx.ext.doctest’,

’sphinx.ext.intersphinx’,

’sphinx.ext.autosummary’,

’sphinx.ext.mathjax’,

’nbsphinx’

]

# Add type of source files

source\_suffix = [‘.rst’,‘.ipynb’,’md’]

# Misc.

When using github pages you need an empty .nojekyll file in same dir as generated html files for html\_theme = 'sphinx\_rtd\_theme' to work.

After building documentation copy everything in build/html to docs/