MIDDLESEX Community College

Tools and Technologies for Tech Writers 2024

Week 7

reStructuredText

Notices

This document was prepared as a handout for the Middlesex Community College Tools and Technologies for Technical Writers class, Winter semester 2024.

Prepared by Zoë Lawson, course instructor.

Contents

reStructuredText resources **4**

Week 7 Homework **5**

Read the Docs quickstart 5
Running Sphinx locally 6
Installing Sphinx 6
Set up a Sphinx project 7
Initial reSt setup 8
Building Sphinx output 8
My first reST project 9

reStructuredText resources

reStructuredText	Markup Syntax and Parser Component of DocUtils https://docutils.sourceforge.io/rst.html The official site with all the official information about reStructuredText. This page has links to various cheat sheets for reStructuredText markup. Or you can go straight to the Quick Reference
Sphinx	https://www.sphinx-doc.org/en/master/index.html An open source tool set for producing output from reSturcturedText. You can install this locally. Also check out the themes gallery: https://sphinx-themes.org
Read the Docs	https://readthedocs.org A free site for hosting Sphinx projects. You can connect a GitHub repository to this site and it will automagically build your output for you.
Online reStructuredText editor	https://www.tutorialspoint.com/online_restructure_editor.php Useful real-time editor/tester of your reST content. Great for experimenting to see if your markup is doing what you expect.

Week 7 Homework

Document a thing using ReST.

Remember to include the following:

- Different levels of headings
- Multiple paragraphs
- Ordered and unordered lists (preferably with some additional paragraphs that are indented properly)
- An image
- A table
- Inline formatting and links

While you can just do some gobbledy-gook random junk, like I do during live demos in class, remember that every homework assignment can be a potential portfolio piece. Yes, you can just hit the bare minimum to prove you've tried to do the homework, or you can look at it as a chance to flush out a budding portfolio.

If you only make one ReST file, just add the file (named with your initials or your name) in the Week 7 homework folder.

If you make a bunch of ReST files, or if you do the Extra, Extra Credit, make a new folder in the Homework folder and provide reStructuredText file(s). I can generate output for you.

All I ask for is a single text file full of reST markup. However, you can explore further if you so choose.

Extra Credit

Make an account on https://readthedocs.org, connect it to a new GitHub repo you make with your reST homework in it, and send me the URL to your generated output.

See Read the Docs quickstart on page 5.

Extra Extra Credit

Transform your content yourself using Sphinx locally.

This involves using the command line, and getting dirty with making software work. This is non-trivial and super technical. This is not for the faint of heart. However, if you want to get into the deep, dark recesses of making content go, it can be fun.

See Running Sphinx locally on page 6.

Read the Docs quickstart

These instructions are quick, but should help you get started, if you want to try and use the automation built in with Read the Docs.

This was last updated when I put this together last year, steps may have changed, so follow the current tutorial.

In reality, you should just go read their tutorial: https://docs.readthedocs.io/en/stable/tutorial/

1. Go to https://readthedocs.org/ and **Log In**.
2. Scroll down and click **Sign in with GitHub**.

   This will automagically connect your GitHub account to Read the Docs, which makes life easier.
3. Follow the buttons to fork their template, or just go to https://github.com/readthedocs/tutorial-template/ and click **Fork**.
4. Once you're in the fork, click **Code** and select **Open in GitHub Desktop**.
5. Follow the prompts as you did when you forked the class repository, however, you do *not* want to contribute back to the parent project. Select **For my own purposes**.
6. Once you have the files locally, you can make your own reStructuredText files and update the `index.rst` in the `docs/source` folder.
7. Commit and push the files to your fork of the Read the Docs template.
8. Go to your Read the Docs project page, such as https://readthedocs.org/projects/mcc-demo-2022/.
9. If the build didn't start automatically, click **Build version**.

The content should build.

I believe there is a delay between when the build says it is complete and when the files actually appear. Tonight I was complaining that I made changes and they weren't working...I was just impatient. I'd wait at least ten minutes before believing your changes didn't work.

Running Sphinx locally

These instructions are not exhaustive, and are probably wrong for a Mac, but they're how I got up and running this week.

Installing Sphinx

These instructions are for Windows. You'll need to adapt for a Mac.

1. Install Python.

   You can check to see if you have Python installed by running one of the following commands:

   ```
   python --version
   python -v
   py --version
   py -v
   ```

   Most web pages say `python --version`, but `py` worked for me with Python 3.10.

   a. Download the current Python installer for your operating system from https://www.python.org/downloads/.

b. Run the installer.
   Python should now be installed on your system. On Windows, it seems to install in `C:\Users\<your user name>\AppData\Local\Programs\Python` by default.

Check the Python version. You may need to start a new command window to pick up the installation.

Note: Python was not added to my path. Therefore, I am actually running `C:\Users\<my User Name>\AppData\Local\Programs\Python\Python311\python.exe` everytime it says `python`.

2. Confirm you have `pip` installed.

   `pip` is a Python tool for installing and updating Python packages/utilities/add-ons.

   a. Run `python -m pip -V`.

      This tells Python to run the `pip` module and return the version. The `V` must be capitalized.

      If `pip` is installed, you should receive a message like `pip 22.0.4 from C:\Users\<your user name>\AppData\Local\Programs\Python\Python310\lib\site-packages\pip (python 3.10)`. If you don't, you need to install pip.

   b. If you need to install `pip`, run `python -m ensurepip --upgrade`.

   c. If you need to upgrade `pip`, run `python -m pip install --upgrade pip`.

3. Install Sphinx by running `python -m pip install Sphinx`.

You should now have the tools installed to run Sphinx locally.

Set up a Sphinx project

1. Make sure your command prompt is in the directory where you want to create your project.
2. Run `sphinx-quickstart`.

   All the web pages say just "run `sphinx-quickstart`" and magic should happen. Windows seems to be a bit different. This year, `C:\Users\<my User Name>\AppData\Local\Programs\Python\Python311\Scripts\sphinx-quickstart.exe` worked for me. Last year, `python -m sphinx.cmd.quickstart` worked for me. You could also try `python -m sphinx-quickstart`.

   The quickstart prompts you to answer questions about your projects.
3. Say `y` to separating build and source.
4. Provide a project name.
5. Provide the author's name.

   In a real project, that probably would be the company or department name.
6. Provide a release number.

   For class, we don't care. For a work project it would matter a lot.
7. Accept the default of `en` for English source.

I now have the following folder structure:

```
build        <-- A folder that will contain my output
source        <-- A folder where my .rst files will go.
    _static    <-- A folder that can hold stuff used in template
    _templates    <-- A folder that holds more layout files
    conf.py    <-- The config file where I can set my theme, etc.
```

```
    index.rst    <-- Here's the start of my TOC
make.bat        <-- This is the command I run to make output
makefile        <-- Text file I can edit to configure the build
```

Initial reSt setup

For a real project, you'd be writing a lot of topics. This is just to get started.

1. Create at least two reStructuredText files in the `source` directory of your Sphinx project.
2. Edit `source/index.rst` and add your files to the `toctree` directive

   All the files must be indented using spaces. Do not use tabs. All the files must be aligned the same.

   This `toctree` directive works:

   ```
   .. toctree::

      file1
      file2
      file3
   ```

   This one does not:

   ```
   .. toctree::

      file1
         file2
      file3
   ```

You are now ready to try building.

Building Sphinx output

It should be as simple as running `make.bat html`.

Unfortunately, on Windows, it doesn't seem to be. I'm hoping on UNIX, `sphinx-build` will work.

1. Set the Windows Environment Variable SPHINXBUILD to the actual command you need to run.

   The Sphinx documentation says you need to run `sphinx-build`. This doesn't seem to be available on Windows.

   Last year, `SET SPHINXBUILD=python -m sphinx.cmd.build` worked for me.

   You can run that in the command window before you build, or you can edit `make.bat` and add it before the `if "%SPHINXBUILD%"` line. This year, I added the following:

   ```
   set SPHINXBUILD=C:\Users\<MyUserName>\AppData\Local\Programs\Python
   \Python311\Scripts\sphinx-build.exe
   ```

   You could also edit the System Environment Variables. These are available from the Control Panel.
2. Run `make.bat html`.

If all goes well, your output will be in the `build/html` folder.

Note: Make sure you open the `index.html` file in the `build/html` folder. The same file in the `build` folder doesn't open the correct frameset. The Contents are probably missing.

If you want to have fun, explore the Themes gallery: https://sphinx-themes.org.

My first reST project

These were my adventures in 2020. I had never used reST until I started preparing this class.

Do not follow these instructions today. Go to https://docs.readthedocs.io/en/stable/tutorial/index.html instead.

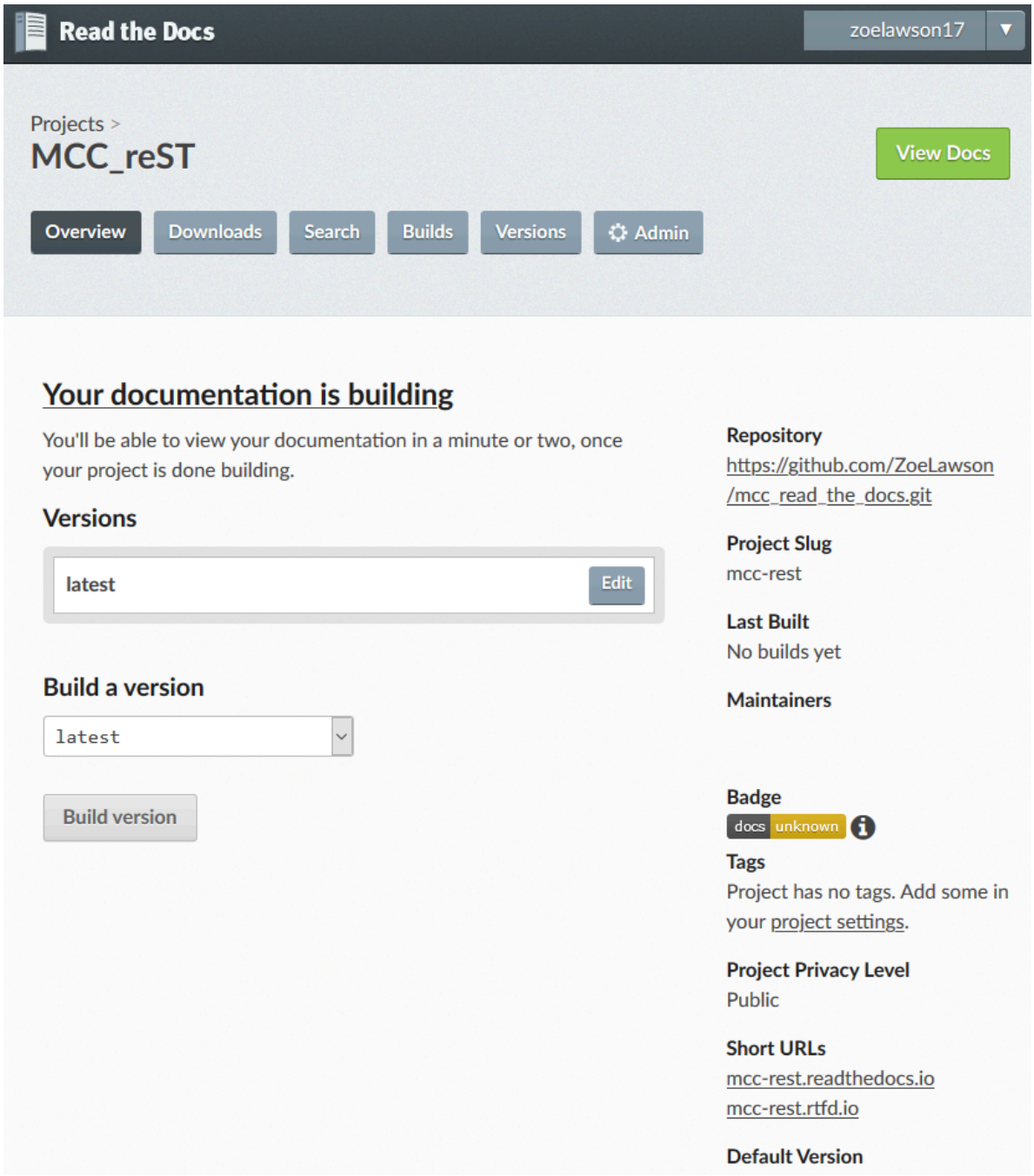
I had tried to work with someone else's reST project in the past, so I had Python and Sphinx installed.

Python 3: https://www.python.org/downloads/

To install Sphinx: https://www.sphinx-doc.org/en/master/usage/installation.html

1. I made an account on https://readthedocs.org/.
2. I made a new public GitHub repository, https://github.com/ZoeLawson/mcc_read_the_docs.
3. I cloned my new mcc_read_the_docs locally and used `sphinx-quickstart` to start a new Sphinx project locally. I just accepted all defaults.
4. I made a new reST file and added it to the `index.rst` file.
5. I committed and pushed my changes to GitHub.
6. I made a new project on readthedocs and connected it to the new repo.

Read the docs automatically builds the content. Once the build completes, I can go and see the output: https://mcc-rest.readthedocs.io/en/latest/index.html.


Read the Docs
zoelawson17
Projects >
MCC_reST
View Docs
Overview
Downloads
Search
Builds
Versions
Admin
Your documentation is building
You'll be able to view your documentation in a minute or two, once your project is done building.
Versions
latest
Edit
Build a version
latest
Build version
Repository
https://github.com/ZoeLawson/mcc_read_the_docs.git
Project Slug
mcc-rest
Last Built
No builds yet
Maintainers
Badge
docs unknown
Tags
Project has no tags. Add some in your project settings.
Project Privacy Level
Public
Short URLs
mcc-rest.readthedocs.io
mcc-rest.rtfd.io
Default Version