Building Jupyter Books Locally on Windows

August 28, 2024

1 Introduction

This document serves as a step-by-step guide on how to locally build Jupyter Books on Windows. It aims to streamline collaboration on projects using Jupyter Books by enabling contributors to preview their contributions locally before submitting a pull request on GitHub.

2 Prerequisites

There are a few prerequisites to following this tutorial.

2.1 Python

Python must be installed, and a path must be set up (note: avoid version 3.8, as it may cause issues). Python downloads are available here for those who do not have it installed. A path should be set up automatically by the Python installer. If this is not the case, please refer to section 2.3.

2.2 Pip

Pip, a package installer for Python, must be installed to facilitate the installation of Jupyter Books. Pip is installed automatically when you install Python. You can check the status of Pip by running the following command in a terminal:

```
C:> py -m pip --version
```

This should output something like:

```
pip X.Y.Z from ... (python 3.N.N)
```

If Pip is not installed for some reason, information on how to do this can be found here. Again, a path should be automatically set up but if this is not the case please refer to section 2.3.

2.3 Step-by-Step Guide to PATHing

When a pathing issue arises there is a relatively simple fix - manually adding the paths. The method for doing this will vary system by system due to file layouts but the tutorial below outlines a typical example for adding PIP to PATH on Windows:

1. **Locate the Pip Directory:**

   - Find the exact location of the `pip` executable. It's usually in the `Scripts` folder within your Python installation directory.

   - The path is often something like:

     ```
     C:\Users\YourUsername\AppData\Local\Programs\Python\PythonXX\
         Scripts\
     ```

   - Replace `YourUsername` with your actual username and `PythonXX` with your Python version (e.g., `Python39` for Python 3.9).

2. **Add the Path to Environment Variables:**

- Open System Properties, locate the **Path** variable in the Environment Variables window, and select **Edit**.

- In the **Edit Environment Variable** dialog, click **New** and add the path to your `Scripts` directory:

```
C :\ Users \ YourUsername \ AppData \ Local \ Programs \ Python \ PythonXX \
    Scripts \
```

- You might also want to add the main Python directory:

```
C :\ Users \ YourUsername \ AppData \ Local \ Programs \ Python \ PythonXX \
```

for the `python` command to be recognized.

3. **Apply and Close:**

- Click **OK** to close all windows and save your changes.

4. **Restart the Command Prompt:**

- Close any open Command Prompt or terminal windows and reopen them. This ensures the changes take effect.

5. **Test Pip:**

- Now, in the Command Prompt, type:

```
pip -- version
```

- If everything is set up correctly, you should see the pip version information.

3   Locally Building a Book

This section outlines the steps required to build a book locally on a Windows machine. It includes links and code snippets where necessary. Credit should be given to the Jupyter Books community for their extensive documentation, upon which this tutorial is heavily based [1].

Please note that in this tutorial, the book directory is named `mynewbook`. However, you can rename it to any name you prefer, as long as you maintain consistency throughout. The functionality will remain intact.

1. **Install Jupyter Books**

- Use the following code snippet within a Windows terminal to install Jupyter Books:

```
pip install -U jupyter - book
```

2. **Generate a sample book**

- Jupyter Book contains a sample book to help you understand a book's structure. Create a sample book by running the following command:

```
jupyter - book create mynewbook /
```

- This will create a file named 'mynewbook' located in:

```
C :\ Users \ YourUsername \ mynewbook
```

- This directory will contain a config file 'config.yml', a table of contents file 'toc.yml', and the content of the book (e.g., '.md', '.ipynb', etc.).

3. **Editing the sample book**

   - To ensure your own files are built into the book you must add them into the file.

   - Add the files into the 'mynewbook' directory along with any image folders.

   - Edit the 'toc.yml' file to add your new file name under the chapters section.

```
! _toc.yml    ✕

C: > Users > joost > mynewbook > ! _toc.yml
   1    # Table of contents
   2    # Learn more at https://jupyterbook.org/customize/toc.html
   3
   4    format: jb-book
   5    root: intro
   6    chapters:
   7    - file: markdown
   8    - file: notebooks
   9    - file: markdown-notebooks
  10    - file: mynewfile
  11
```

4. **Building the edited sample book**

   - To build the sample book so you can view it locally use the following command:

```
jupyter-book create mynewbook/
```

   - The terminal should now output a URL which you can copy/paste into a browser to view the book in HTML.

5. **Making further edits to the book**

   - To edit the sample book simply add any new files into the directory and add them to the 'toc.yml' file.

   - To rebuild the sample book so you can view it locally use the following command:

```
jupyter-book build --all mynewbook/
```

   - The terminal should now output a URL which you can copy/paste into a browser to view the book in HTML.

4   Final Notes

This tutorial allows you to place your markdown files into a test book, meaning you can test an individual file without having to build the entire textbook. This, of course, is not always ideal since you may want to build the full book. In cases such as these, it is best to consult the Jupyter Books documentation in full since this guide is just designed to be a quick start to testing individual pages for review purposes (albeit in a bit of a quick and dirty manner). The full documentation can be found here.

If there are any issues, please don't hesitate to contact me at j.hubbard@se21.qmul.ac.uk.

5   References

[1] Jupyter Book Community. *Jupyter Book Documentation*. `https://jupyterbook.org/en/stable/start/your-first-book.html`. Accessed: 2024-08-28. 2023.

[2] Python Software Foundation. *Getting Started with Pip*. Accessed: 2024-08-20. 2024. URL: `https://pip.pypa.io/en/stable/getting-started/`.

[3] Python Software Foundation. *Python Releases for Windows*. Accessed: 2024-08-20. 2024. URL: `https://www.python.org/downloads/`.