

DFFML: Intuitive and Accessible Documentation Editing

About Me:

1. Name: Claire Follett
2. Github handle: Clairefollett
3. University: Clatsop Community College, Oregon State University
4. Time Zone: Pacific Time (UTC - 8:00)
5. Email: chlaw101@gmail.com

Project Information:

Sub-Org Name:

Data Flow Facilitator for Machine Learning (DFFML).

Project Abstract:

Documentation is not written or updated as much as it should be because the overhead is too great. This project aims on integrating existing tooling to provide an intuitive and accessible process to write and edit documentation. The implementation of this solution can be broken down into two phases:

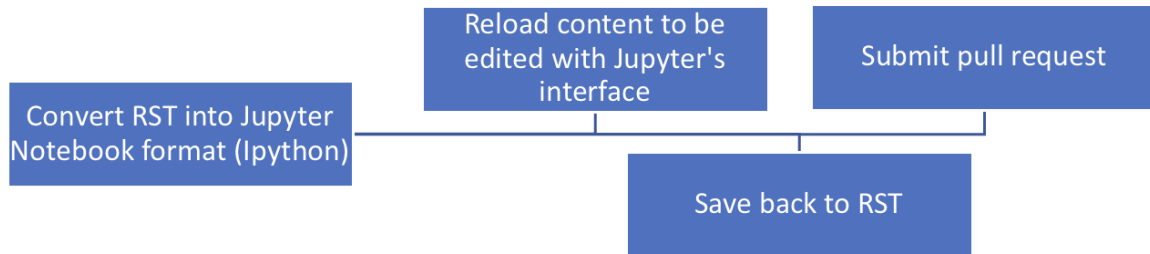
1. Hotswapping page content for code editor
2. Generating pull requests based on content delta

Detailed Description

The purpose of DFFML is to make processes more automated and simpler. One of the areas that most projects struggle with in this sense is documentation and communication between the team. A lot of developers struggle with writing thorough and well-organized documentation, and a lot of projects lack a detailed process of how to do so in the most efficient and in-depth way.

We can accomplish this by having an automated process when a developer wants to contribute or add changes to a project, making it easier to communicate changes

through a simple form or document, then transforming that document into a nicely edited template/pull request. The ideal flow would look something like this:



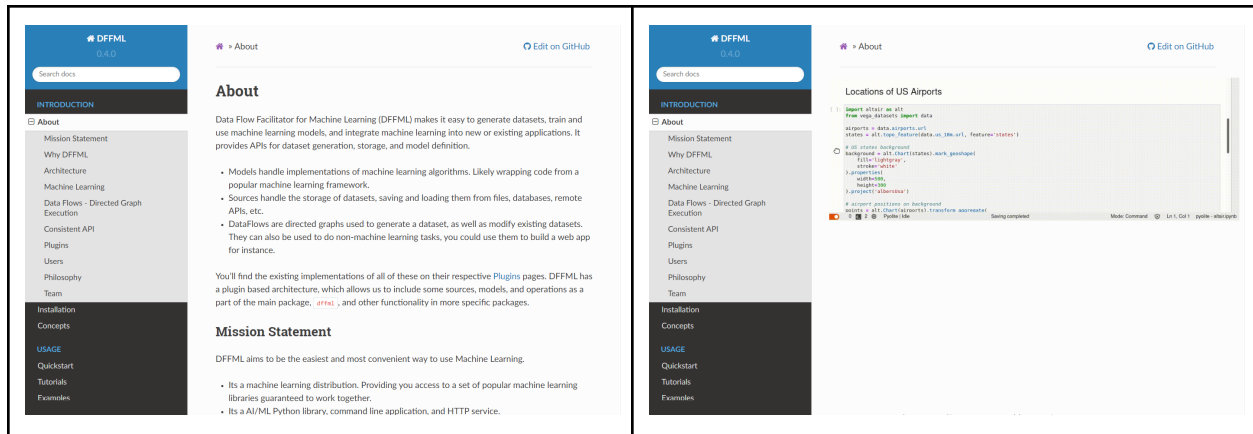
There are more than a few projects/plugins out there that accomplish one or more of the steps above, but none that make the process straightforward or can integrate all the steps together.

Convert rst into jupyter notebook format: There have been multiple projects/tools that accomplish this.

Example of similar projects:

- a. Vertopal - Free Online Converter:
<https://www.vertopal.com/en/convert/rst-to-ipynb>
- b. Rst2ipynb - Convert standalone reStructured text to Jupyter notebook files:
<https://github.com/nthiery/rst-to-ipynb>
They achieve this by converting the file to markdown using pandoc (<https://pandoc.org/>) and then to a Jupyter notebook using Notedown (<https://github.com/aaren/notedown/>).
- c. A sphinx extension for making Jupyter Notebooks:
<https://github.com/QuantEcon/sphinxcontrib-jupyter>

Reload content to be edited with jupyter interface: Hotswap page content for code editor:



Saves back to RST: There have been a couple of different projects/tools that accomplish this. I came across a tool that makes this process pretty straightforward. It's the nbconvert tool, which can be found at: <https://nbconvert.readthedocs.io/en/latest/>. The nbconvert tool will produce HTML, but not rst, due to missing transformation to docutils intermediate representation. This is a major gap that needs to be addressed and filled, in order to generate a pull request that can be accepted as an underlying restructured text file.

Submit pull request

Weekly Timeline

- **Community Bonding:**
 - Keep in touch with GSOC mentors, contribute to issues on Github.
 - Familiarize myself with workflow practices on DFFML, and the code base.
 - Deploy Jupyterlite and figure out where interfaces are for opening and saving files.
- **Week 1 (June 13th):**
 - Start working on the integration tool to convert rst files to Jupyter Notebook format.
- **Week 2 (June 20th):**
 - Continue implementation of conversion from rst files to Jupyter Notebook format.

- **Week 3 (June 27th):**
 - Continue work from week 1 and 2. Work on figuring out how to open and save files within Jupyter.
- **Week 4 (July 4th):**
 - This week is open for any unforeseen obstacles or other issues that come up. Continue with the work from weeks 1, 2, and 3.
- **Week 5 (July 11th):**
 - Touch base with mentors about project trajectory.
 - Wrap-up work from the first few weeks.
- **Week 6 (July 18th):**
 - Begin work on writing JavaScript that will convert Ipython (Jupyter Notebook format - JSON) back into reStructured text.
- **Week 7 (July 25th):**
 - Continue working on implementation of conversion from Ipython to rst.
- **Week 8 (August 1st):**
 - Continue work from weeks 6 and 7.
- **Week 9 (August 8th):**
 - Finish implementing conversion from Ipython back to rst.
- **Week 10 (August 15th):**
 - Discuss with mentors any complications that arise. Fix and work on any issues that arise.
 - Begin work on submitting automated pull requests from the rst files.
- **Week 11 (August 22nd):**
 - Wrap up work for the automated pull requests. Fix any unforeseen issues.
- **Week 12 (August 29th):**
 - Finish any pending work and documentation.

Other Commitments

I will be graduating with my Associate of Science - Oregon Transfer Degree in Computer Science on June 17th, 2022. I begin my studies at Oregon State University in the Fall of 2022, at the end of September. I have no official

commitments between June and September of 2022. I plan to spend 8 hours, 5 days a week on this project.