Application: Jupyter Notebook maintenance and community support

EOSS5: Essential Open Source Software for Science (Cycle 5)

Summary

ID: EOSS5-0000000267
Last submitted: Apr 17 2022 05:46 PM (CEST)

1. Applicant Details

Completed - Apr 15 2022

1. Applicant Details

Complete the following information for the Applicant (required)

The information entered should be for the individual submitting the application who will act as the main person responsible for the application and as its point of contact. To edit your name or email, navigate to Account Information by clicking your name in the upper right corner.

Name: Frederic Collonval

Add your home institution, company, or organization. This does not need to be the organization to which a grant would ultimately be awarded, if selected for funding.

Institution/Affiliation	QuantStack

2. Proposal Details

Completed - Apr 17 2022

2. Proposal Details

a. Proposal Title: Jupyter Notebook maintenance and community support

To edit your proposal title, navigate to the main page; click on the three dots to the right of the application title; and select Rename from the dropdown menu. Proposal title is limited to 60 characters including spaces.

b. Amount Requested

Enter requested budget in USD, including indirect costs. This number should be between $100k and $400k over a two year period. Enter whole numbers only (no dollar signs, commas, or cents).

400000

c. Proposal Summary/Scope of Work

Provide a short summary of the work being proposed (maximum of 500 words)

Jupyter notebooks are very popular in all areas of technical computing, from life sciences to data engineering, robotics, and geo-sciences.

Two user interfaces are proposed by the Jupyter project to manipulate notebooks:

1. The "Classic Notebook", a single-document full-page editor, which is not actively developed anymore,
2. JupyterLab, which is built upon more modern foundations, and is receiving the major part of the development effort.

As many users continue using the classic notebook interface, it was decided by the Jupyter steering council (cf. Jupyter Enhancement Proposal #79 [1]) that the next major release of the classic notebook, notebook v7, will be built from JupyterLab components. This will allow the classic notebook to benefit from the improvement brought to JupyterLab in many areas such as accessibility, internationalization, collaborative editing, visual debugging, language server protocol support, and much more. Key objectives of this migration are:

Retaining the document-centric user experience of the classic notebook and minimizing the changes to the user interface

Porting the most popular classic notebook extensions to JupyterLab and notebook v7. Indeed, the extension system of this major release will be completely different from earlier ones.

While this migration will directly impact millions of users and is key to the future of the Jupyter project, it is currently being done by volunteers in their free time. There are major challenges ahead, to ensure the best possible migration for end-users, extension authors, and engineers supporting Jupyter deployment at universities.

This proposal is focused on supporting this migration, with a strong focus on documentation across the Jupyter project.

Documentation

- Refactoring of the existing Jupyter documentation:
-- Centralizing documentation currently spread across multiple websites
-- Improving the documentation for new contributors and plugin developers starting from scratch or migrating from the old classical notebook extension system.
-- Reorganizing the documentation following the four pillars approaches: tutorials, how-to guides, technical reference and explanation.
- Providing an internationalized documentation website
-- The current tools do support internationalization. The goal will be to set up automation scripts for the internationalized documentation website publication and the documentation translation through a third-party service (like the one used for the interface translation).
-- Create a library of helpers to generate documentation screencasts automatically to keep them up to date and use the translated UI for internationalized documentation. The library will be released as a separated project to increase its impact.
- Providing better guided tours for the user interface
-- A JupyterLab extension to create guided tours does exist. The goal will be to make it a core feature of the JupyterLab project, and to make it easier to create new interface tours without coding.

Code base

- Support the community extensions authors to migrate their code
- Address issues related to backward inconsistencies in the new major classical notebook version.

- Reduce the issues backlog focusing on bug fixes and features requested by the biomedical community based on the 2021 survey [2]; at the time of this proposal writing, the document generation features and easier extension creation are the candidates [3].

[1] https://jupyter.org/enhancement-proposals/79-notebook-v7/notebook-v7.html
[2] https://github.com/jupyter/surveys/tree/master/surveys/2020-12-jupyter-survey
[3] https://github.com/jupyterlab/team-compass/issues/121#issuecomment-782913362

d. Value to Biomedical Users

Described the expected value the proposed work to the biomedical research community (maximum of 250 words)

JupyterLab and Notebook are primary tools used for biomedical science education, research, and manufacturing. Their interface provides a set of tools allowing users to work with code, data and visualization in a single document. Jupyter’s growth in schools, universities and laboratories speak to the value of these technologies in science.
The biomedical community has for example created extensions such as itkwidgets [1], ipycytoscape [2] or jupyterlab-citation-manager [3]. Research results (e.g. [4]) and tutorials (e.g [5]) are also shared using Jupyter Notebook document.

Moving to newer technologies for the classical notebook is required to ensure the maintenance cost stays manageable and the tool will keep up with new browser features and requirements.
Keep improving the tool, focusing on research code documentation, to facilitate sharing research results and to foster an understanding of new discoveries.

[1] https://github.com/InsightSoftwareConsortium/itkwidgets
[2] https://github.com/cytoscape/ipycytoscape
[3] https://github.com/krassowski/jupyterlab-citation-manager
[4] https://github.com/SFB-ELAINE/Biomedical-Jupyter-Examples
[5] https://goodboychan.github.io/python/datacamp/deep_learning/vision/2020/08/13/01-Exploration-in-Biomedical-Image-Analysis.html

e. Open Source Software Projects

Number of software projects are involved in your proposal (maximum of five):

5

Complete the table with the following information for each software project. If there is no homepage URL, re-enter the main code repository URL.

	Software project name	Main code repository URL	Homepage URL
1	JupyterLab	https://github.com/jupyterlab/jupyterlab	https://jupyterlab.readthedocs.io/
2	Notebook	https://github.com/jupyter/notebook/	https://jupyter-notebook.readthedocs.io/
3	Lumino	https://github.com/jupyterlab/lumino/	https://lumino.readthedocs.io/
4	JupyterLab server	https://github.com/jupyterlab/jupyterlab_server	https://jupyterlab-server.readthedocs.io/
5	Jupyter server	https://github.com/jupyter-server/jupyter_server/	https://jupyter-server.readthedocs.io/

f. Landscape Analysis

Briefly describe the other software tools (either proprietary or open source) that the audience for this proposal primarily uses. How do the software project(s) in this proposal compare to these other tools in terms of user base size, usage, and maturity? How do existing tools and the project(s) in this proposal interact? (maximum of 250 words)

Other known desktop tools used by the scientific community to document code and visualized results are RStudio, JetBrains PyCharm/DataSpell, VSCode or nteract. Some tools are also available from online services like Google Colab(oratory) or CoCalc. Out of those three, RStudio is the most used tool in the scientific biomedical community as it supports the R language (very popular for statistical analysis) and is well established.

Lately VSCode has invested lots of time integrating Jupyter notebook within the nowadays most popular open-source Integrated Development Editor. So their potential user base is bigger than Jupyter. But having a broader audience the extensibility of the tool is more constrained than the more dedicated Jupyter tools.

nteract main goal is to provide a non-extensible notebook editor with a limited set of features compare to Jupyter tools.

Except nteract and VSCode, the other tools are being developed by companies that provide paid versions. That financial source helps address bug fixes and reduce the backlog compared to the fully open source community-driven Jupyter tools. VSCode being a Microsoft product benefits too of a strong financial support as part of the tool is reused in commercial products.

Jupyter tools are managed by the Project Jupyter organization. Compare to the other tools (except nteract), it is the only entity with an open governance [1] allowing any entities to be part of the decisions and development roadmap.

[1] https://jupyter.org/governance/intro.html

g. Category

Choose the two categories that best describe the software project(s) audience.

	Category
Category 1	Machine learning and data analysis
Category 2	Bioinformatics

h. Previous CZI Funding

Did you previously apply for funding for this or a related proposal under the CZI EOSS program?	No
Have you previously received funding for this proposal under the CZI EOSS program?	No

Completed - Apr 15 2022