

Application: Jupyter Notebook maintenance and community support

Frederic Collonval - frederic.collonval@quantstack.net
EOSS5: Essential Open Source Software for Science (Cycle 5)

Summary

ID: EOSS5-0000000267

Last submitted: Jun 2 2022 10:40 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

Email: frederic.collonval@quantstack.net

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/kkrassowski/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/notebook-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

3. Equal Opportunity & Diversity

Completed - Apr 15 2022

Equal Opportunity & Diversity

CZI Science supports the science and technology that will make it possible to cure, prevent, or manage all diseases by the end of this century. Everyone is affected by disease, yet different communities are affected by or experience disease in different ways. Moreover, due to systemic barriers, the scientific enterprise itself is not a place where all voices and talents thrive. We believe the strongest scientific teams — encompassing ourselves, our grantees, and our partners — incorporate a wide range of backgrounds, lived experiences, and perspectives that guide them to the most important unsolved problems. To enable our work, we incorporate diverse perspectives into our strategy and processes, and we also seek to empower community partners to engage in science.

We request demographic information associated with applications submitted to CZI in response to our open calls. This information helps us learn from the RFA process, as well as improve our strategies to help ensure members of underrepresented or marginalized groups in science are aware of and able to apply to CZI opportunities. **Please note that answering all questions below is voluntary, and demographic information will not be used to make final grant funding decisions.** All responses will be shared

6. Institutional Approval Form (required):

Upload as a single PDF. This [form](#) should be reviewed and signed by a person authorized to sign on behalf of your organization agreeing to the stated institutional and investigator requirements and commitments on data, resource sharing, and publication policies, as well as endorsing/verifying your application materials and confirming their ability to receive funding for the proposal. In the event of an award, all funds will be awarded to the applicant institution as the prime institution, and the applicant institution will be responsible for ensuring compliance of all of the terms, including compliance of all partners/subcontract institutions. **These policies are non-negotiable so this form should only be signed if the organization is able to comply with the terms as stated.** While CZI does not require sign-off by all of your partner institutions, please refer to what your institution requires. **Note: digital signatures are permitted as long as the document is not encrypted or password-protected.**

[Cycle 5 Institutional Approval Form - signed.pdf](#)

Filename: Cycle 5 Institutional Approval Form - signed.pdf **Size:** 2.7 MB

3. Proposal Details

Completed - Jun 2 2022

Proposal Details

*Please complete the following proposal information. **All sections are required.***

1. Proposal Title: Jupyter Notebook maintenance and community support

2. Previous Funding

Did you previously apply for funding for this or a related proposal under the CZI EOSS program?

No

3. Proposal Purpose:

Describe the purpose of the proposal in one sentence (maximum of 200 characters including spaces). Example: To develop a comprehensive, validated atlas of the human kidney at single-cell resolution open to the entire scientific and clinical community.

Improve Jupyter Notebook documentation to ease users and contributors understand the software and support the community usage by reducing the issues and requested features backlog.

4. Amount Requested:

Enter the amount requested per year between \$50,000 USD and \$200,000 USD per year, including indirect costs), as well as the total budget requested for all years in USD (between \$100,000 USD and \$400,000 USD total, including indirect costs). These numbers should match those described in the Budget Description to follow. Enter whole numbers only (no dollar signs, commas, or cents)

Year One	200000
Year Two	200000
Total All Years	400000

5. Proposal Summary:

Provide a short summary of the application (maximum of 500 words) (auto-filled from LOI; update if needed)

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's primary goal is to support 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 separate 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

- 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>

6. Work Plan:

A description of the proposed work for which funding is being requested, including resources the applicants will provide that are not part of the requested funding. For software development-related work (e.g., engineering, product design, user research), specify how the work fits into the existing software project roadmap. For community outreach related activities (e.g., sprints, training), specify how these activities will be organized, the target audience, and expected outcomes (maximum of 750 words)

By the end of the summer 2022, the new major versions of the supported frontends for Jupyter notebook, JupyterLab (v4) and Notebook (v7), will be released jointly. This will be the first major release of the Classic Notebook to be based on the modern components built for JupyterLab. The main features of Notebook v6 will be supported in Notebook v7 (and new ones will be added). While the user interface will be very similar, extensions will need to be ported to the JupyterLab extension system.

To help with that migration, two axes are developed in this proposal: improving the documentation and helping users migrate to the new version. These two axes will be worked on in parallel by a senior software developer for 20h/week and a software developer for 20h/week.

The first task will be to organize weekly virtual office hours during which developers will receive help to migrate their code in peer-programming sessions and mentoring by an experienced Jupyter contributor. That hour will also be a good opportunity to gather pain points to be lifted in the documentation or in the code base.

On the documentation front, the first task will be to set up a centralized documentation website bringing together the documentation of the different projects involved in the final user interface; namely jupyter-

server, jupyterlab-server, lumino, notebook and jupyterlab. Once the new structure is in place, we will set up internationalization for the documentation website. It will require setting up automation scripts for publishing the reference English documentation to a third-party service for translators. Then other scripts will download those translations and publish the translated documentation.

Once the new infrastructure is in place, the documentation for developers will be re-organized following the four pillars approaches: tutorials, how-to guides, technical reference, and explanation. From there, additional documentation will be written focusing on how to create extensions, on how to achieve common tasks (such as adding visual elements to a cell, adding a side panel,...) and addressing pain points identified in the office hours.

Guided tours of the interface will also be part of the additional documentation to help users understand the link between code and visual elements.

In parallel to the writing task, a library of helpers to generate screencasts of the user interface as part of the continuous integration workflows will be created and published as a separate project. The library will take inspiration from [1] but applying the concept on the integration test tool used by Jupyter frontends; i.e. Playwright [2]. Once this tool is available documentation screencast scenarii will be added to the CI as well as the needed scripts to publish them in the documentation website.

On the development side, as the grant funding will start a few months after the new release is published, the initial focus will be on resolving reported bugs. Once the stream of new bugs will decrease, we will start addressing the issue backlog focusing on the JupyterLab issue tracker and the Classic Notebook issue tracker. In parallel to that continuous improvement, development of the two new major features standing out from the latest user survey [3] will be addressed:

Making extension development easier. The experimental jupyterlab-plugin-playground [4] packaged will be finalized to allow quick implementation of extensions. Some upstream work will be carried out to allow dynamic loading (and unloading) of frontend plugins and of server extensions.

Create educational content and document research, the executable books organization [5] has created great tools based on Jupyter notebook for that purpose. But the integration in a Jupyter notebook frontend is not yet available. There is an experimental jupyterlab-myst extension [6]. This feature will therefore be fulfilled by building on that extension in particular helping with needed changes in core Jupyter projects.

Finally if time allows it, actions to ease the code maintenance and reduce the technical debts will be carried out too. At the time of writing this proposal, two large actions are envisioned: expand the usage

of a components toolkit [7] and start working on the next major version of the underlying Lumino widget library [8]. The first point will reduce the technical debts and ease extension developers' work. And the second will focus on moving away from our custom web widget class and embrace web standards to reduce the learning curve and ease integration of Jupyter elements with other web frameworks.

[1]: <https://github.com/bahmutov/cypress-movie/>

[2]: <https://playwright.dev/>

[3]: <https://github.com/jupyterlab/team-compass/issues/121#issuecomment-782913362>

[4]: <https://github.com/jupyterlab/jupyterlab-plugin-playground>

[5]: <https://jupyterbook.org/en/stable/intro.html>

[6]: <https://github.com/executablebooks/jupyterlab-myst>

[7]: <https://github.com/jupyterlab/team-compass/issues/143>

[8]: <https://github.com/jupyterlab/lumino/issues/233>

7. Milestones and Deliverables:

List expected milestones and deliverables, and their expected timeline. Be specific and include where possible any goals for metrics the software project(s) are expected to reach upon completion of the grant. Please use a third-person voice (maximum of 500 words).

Timeline is expressed in T0 (project launch) + number of months; e.g. T0+3M means 3 months after project launch.

Documentation

Milestone 1: Reorganize documentation website to gather from multiple sources (T0+2M)

When this milestone is completed, users will be able to access the documentation for all components involved in JupyterLab and the Classic Notebook within a single domain (jupyter-server, jupyterlab-server, lumino, notebook and jupyterlab).

Milestone 2: Set up internationalization (T0+4M)

This milestone will be completed when the documentation for jupyter-server, jupyterlab-server, lumino,

notebook and jupyterlab will be sent by automatic scripts to a translation third-party service and the translation will be uploaded to the internationalized documentation website.

Milestone 3: Restructure the developer documentation (T0+6M)

At completion, the developer documentation will be structured in four parts: tutorials, how-to guides, technical reference and explanation.

Milestone 4: Develop screencast tool (T0+9M)

This milestone will be concluded by the publication of a library for Playwright allowing to record documentation videos from integration tests adding cursor tracking and comments (as video subtitles) and the publication of the documentation website with at least 50% of the videos (or 26 videos) generated using that library.

Milestone 5: Provide guided tours for the user interface (T0+12M)

To fulfill this milestone, at least two guided tours will be added. Tour ideas: general interface, notebook interface, table of contents usage and debugger usage.

Milestone 6: Write documentation (T0+24M)

This milestone will be fulfilled by writing at least 1 new tutorial and 5 how-to guides.

Code

Milestone 1: Set up a weekly contribution office hour to help the community migrating their code (T0+1M)

This milestone will be complete when a recurrent 1hour weekly meeting will be set on the Jupyter community calendar and advertisement for it will be done on the Jupyter forum.

Milestone 2: Reduce the issues backlog and technical debt (T0+24M)

This milestone will be fulfilled when open issues on <https://github.com/jupyterlab/jupyterlab> will be 40% lower than now (or less than 1293 issues).

Milestone 4: Addressed user-survey key new features (T0+24M)

Milestone 4.1: Ease extension creation (T0+12M)

This milestone will be complete when Lumino plugins and server handlers can be dynamically loaded and unloaded and that mechanism will be integrated in the jupyterlab-plugin-playground extension.

Milestone 4.2: Create documentation content for science in JupyterLab (T0+24M)

This milestone will be complete when it will be possible to edit documents using MyST syntax and to see a preview of it within JupyterLab.

8. Existing Support:

List active and recently completed (previous two calendar years) financial or in-kind support for the software project(s), including duration, total costs in USD, and source of funding. Include any previous funding for these software projects received from CZI outside of the EOSS program (maximum of 250 words).

The Jupyter Project encompasses a large number of subprojects, developed by many teams around the world, here is a list of grants that the Jupyter Project received. None were for direct development of the project described in this proposal.

- 2014-2020: Helmsley Charitable Trust, Moore Foundation, Sloan Foundation – Original Jupyter grant - \$6,000,000 - Grant received by UC Berkeley and Cal Poly
- 2018-2021: Sloan foundation – Enabling Safe Access to Sensitive Data - \$1,680,000 - Grant received by UC Berkeley and Cal Poly
- 2018-2020: Schmidt Futures – Integrated Digital Information System Research- \$2,200,000 – Grant received by Cal Poly
- 2019-2021: Helmsley Charitable Trust – Jupyter Sustainability – \$1,000,000 – grant Received by UC Berkeley and Cal/Poly
- 2019-2022: National Science Foundation – Jupyter meets the Earth: Enabling discovery in geoscience through interactive computing at scale – \$1,700,000 – Grant received by UC Berkeley
- 2019-2020: Chan Zuckerberg Initiative – JupyterHub Contributor in Residence – \$141,000 – Grant received by NumFocus
- 2020-2021: Chan Zuckerberg Initiative – Real-Time Collaboration in Jupyter – \$241,000 – Grant received by QuanSight
- 2021-2023: Chan Zuckerberg Initiative – Inclusive and accessible scientific computing in the Jupyter Ecosystem – \$340,000 – Grant received by QuanSight
- 2021-2023: Chan Zuckerberg Initiative – JupyterHub Community Strategic Lead - \$399,930 - Grant received by NumFOCUS
- 2021-2023: Chan Zuckerberg Initiative – Papyri: Better Documentation for the Scientific Ecosystem in Jupyter - \$191,139 – Grant received by NumFocus
- 2015-Present: Bloomberg, Two Sigma, Amazon, and others – Collaboration with employees. In-kind.
- 2015-Present: DE Shaw, Goldman Sachs, and other companies – Commercial funding for development

9. Landscape Analysis:

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). (auto-filled from LOI; update if needed)

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

10. Value to Biomedical Users:

Describe the expected value of the proposed work to the biomedical research community (maximum of 250 words). (auto-filled from LOI; update if needed)

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

11. Category:

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

Category 1	Machine learning and data analysis
Category 2	Bioinformatics

12. Diversity, Equity, and Inclusion Statement:

Advancing DEI is a [core value](#) for CZI, and we are requesting information on your efforts in this area. Describe any efforts the software project(s) named in this proposal have undertaken to increase diversity, equity, and inclusion with respect to their contributors and audience. Please see [examples](#) from applications funded in previous cycles (maximum of 250 words)

Project Jupyter is committed to creating an inclusive and equitable environment for community members from all walks of life. It has prioritized inclusion by creating an inviting, welcoming, and supporting community culture. So far, efforts to introduce and support marginalized groups have focused on community events and sub-projects such as organizing regular community meetings across several time zones. Also, there have been considerable efforts to onboard new contributors through programs like Outreachy, the Jupyter Contributor in Residence initiative and with support from Cal Poly San Luis Obispo.

Lack of accessibility and standards in the design and development of digital technologies and infrastructure can prevent a substantial number of people with disabilities from achieving social inclusion and digital literacy. Jupyter organization is currently working on improving the tools accessibility thanks to another CZI grant (EOSS4-0000000164) focusing on accessibility and other roadmap items such as using an accessible toolkit [1]. One important recent achievement is the creation of a sub-project focusing on accessibility as part of the Jupyter organization [2].

[1] <https://github.com/jupyterlab/team-compass/issues/143>

[2] <https://github.com/jupyter/accessibility/issues/81>

4. Optional Attachments

Incomplete

Attachments should be uploaded in a combined single PDF. This section can include figures, charts and tables, references for the proposal, or any additional material in support of the proposal (maximum of three pages). Uploading any additional information is optional.

5. CV of Applicant

Completed - Jun 2 2022

Upload in PDF format; include current and recent employment, education history, and references to any major publications, software contributions, or other relevant outputs (maximum of two pages)

[FCollonval CV](#)

6. Budget Description

Completed - Jun 2 2022

Upload in PDF format; budgets can be uploaded in a combined single PDF or one PDF for each software project (one page per software project maximum)

- Description of the costs to be funded by this grant at a high level and in narrative or tabular form, outlining costs for personnel (including names, if known), supplies, equipment, travel, meetings/hackathons/sprints, subcontracts, other costs, and up to 15% indirect costs (excluding equipment and subcontracts).
- Indirect costs are limited to up to 15% of direct costs and are included within the annual budget total. Indirect costs may not be assessed on capital equipment or subcontracts, but subcontractors may include up to 15% indirect costs of their direct costs. **Non-charitable entities must include a clear allocation and explanation for any indirect costs included in a proposed budget.**
- Budget should be requested in US dollars.
- International grantees must use all grant funds exclusively for activities conducted outside the United States of America. Travel expenses to the United States (including round-trip tickets) should not be covered from the requested grant funds.
- Application budgets must reflect the actual needs of the proposal. The Chan Zuckerberg Initiative will work closely with successful applicants to arrive at a mutually acceptable budget after review.

Jupyter-Maintenance-Proposal budget

Filename: Jupyter-Maintenance-Proposal_budget.pdf Size: 77.2 kB

7. Open Source Software Project(s) Details

Completed - Jun 2 2022

Open Source Software Project Details

Provide details and metrics for each open source software project that will be supported by the grant to help us assess its impact and quality.

How many software projects involved in your proposal (up to five)? If multiple software projects are involved, details must be entered for all of them where indicated below.

5

SOFTWARE PROJECT #1

SOFTWARE PROJECT #1: Details:

Complete the following table for Open Source Software Project #1 of your proposal. All URLs should be in the format <https://example.com> and only one primary link should be provided.

Software Project name (required)	JupyterLab
Main code repository (e.g. GitHub URL) (required)	https://github.com/jupyterlab/jupyterlab
Homepage URL (required)	https://jupyterlab.readthedocs.io/
Social media handles (if applicable)	@projectjupyter
Do you or software project key personnel have commit rights to the code repositories for this software project? (required)	Yes
Short description of software project (200 words maximum) (required)	<p>An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.</p> <p>JupyterLab is the next-generation user interface for Project Jupyter offering all the familiar building blocks of the classic Jupyter Notebook (notebook, terminal, text editor, file browser, rich outputs, etc.) in a flexible and powerful user interface.</p>

SOFTWARE PROJECT #1: List of Key Personnel:

Key personnel are people involved in the software project who will be supported by the grant if the application is successful.

Complete the following for the key personnel on the open source software project #1 (up to 5) (required); **enter n/a if any field is not applicable**. Personnel to be hired that have not been identified at this time can be listed in the budget section. You may need to use the scroll bar at the bottom of the table to scroll right to view and to complete all fields. Alternatively, you can tab to move through and complete the fields. **To add another person/row (up to five), click the box at the end of the row.**

--	--	--	--	--	--	--	--	--

1. SOFTWARE PROJECT #1 : What is the software project license?

Permissive license (e.g. BSD 3-Clause, MIT, Apache 2.0)

2. SOFTWARE PROJECT #1 : What is the main programming language?

Other (please specify): Typescript

3. SOFTWARE PROJECT #1: Does the software project have a code of conduct?

https://github.com/jupyterlab/.github/blob/master/CODE_OF_CONDUCT.md

4. SOFTWARE PROJECT #1: Does the software project have end-user documentation?

<https://jupyterlab.readthedocs.io/>

5. SOFTWARE PROJECT #1: Does the software project have an issue tracker?

<https://github.com/jupyterlab/jupyterlab/issues>

6. SOFTWARE PROJECT #1: Does the software project have a community engagement / Q&A forum (self-hosted, on Stack Exchange etc.)?

<https://discourse.jupyter.org/>

7. SOFTWARE PROJECT #1: Does the software project have contribution / coding guidelines?

<https://jupyterlab.readthedocs.io/en/latest/developer/contributing.html>

8. SOFTWARE PROJECT #1: Is there a corresponding package available in a package manager (PyPi, CRAN, etc.)?

<https://pypi.org/project/jupyterlab/> <https://anaconda.org/conda-forge/jupyterlab>

9. SOFTWARE PROJECT #1: Does the software project support continuous integration for testing?

It uses GitHub CI workflows <https://github.com/jupyterlab/jupyterlab/tree/master/.github/workflows>

SOFTWARE PROJECT #1: Metrics- Impact (optional):

*Complete the following for the open source software project #1. **Providing metrics is optional and metrics can be approximate.** For each metric, please provide a source, clarify how the metric was computed, and/or provide any other comments. For monthly metrics, please provide data from the most recent month for which the corresponding metric is available.*

1. SOFTWARE PROJECT #1: Complete the following table. List the number and explanation for each, if needed:

	Number	Comment
Scholarly paper(s) (including preprints) citing or mentioning the software project		
Monthly users, if applicable (based on one or more of the following: monthly downloads from websites, monthly downloads from package managers, monthly unique requests for updates, etc.)	3415381	Source: https://pepy.tech/project/jupyterlab
Software projects that depend on the project	4783	Number of GitHub repositories depending on @jupyterlab/application: https://github.com/jupyterlab/jupyterlab/network/dependents?package_id=UGFja2FnZS00NTk1Njl2MzA%3D
Monthly visitors to project's website, discussion forum (e.g. Stack Overflow)		

2. SOFTWARE PROJECT #1: List of software projects to which key personnel on this proposal are contributing.

(No response)

SOFTWARE PROJECT #2

SOFTWARE PROJECT #2: Details:

Complete the following table for Open Source Software Project #2 of your proposal. All URLs should be in the format <https://example.com> and only one primary link should be provided.

Software Project name (required)	Notebook
Main code repository (e.g. GitHub URL) (required)	https://github.com/jupyter/notebook/
Homepage URL (required)	https://jupyter-notebook.readthedocs.io/
Social media handles (if applicable)	(No response)
Do you or software project key personnel have commit rights to the code repositories for this software project? (required)	Yes
Short description of software project (200 words maximum) (required)	The Jupyter notebook is a web-based notebook environment for interactive computing with a single-document interface.

SOFTWARE PROJECT #2: List of Key Personnel:

Key personnel are people involved in the software project who will be supported by the grant if the application is successful.

Complete the following for the key personnel on the open source software project #2 (up to 5) (required); **enter n/a if any field is not applicable**. Personnel to be hired that have not been identified at this time can be listed in the budget section. You may need to use the scroll bar at the bottom of the table to scroll right to view and to complete all fields. Alternatively, you can tab to move through and complete the fields. **To add another person/row (up to five), click the box at the end of the row.**

	First name	Last name	Email address	Current employer /Affiliation	Job title	Developer username if applicable	Country of Residence	Add another person/row

2. SOFTWARE PROJECT #2 : What is the main programming language?

Other (please specify): Typescript

3. SOFTWARE PROJECT #2: Does the software project have a code of conduct?

https://github.com/jupyter/governance/blob/master/conduct/code_of_conduct.md

4. SOFTWARE PROJECT #2: Does the software project have end-user documentation?

<https://jupyter-notebook.readthedocs.io/>

5. SOFTWARE PROJECT #2 : Does the software project have an issue tracker?

<https://github.com/jupyter/notebook/issues>

6. SOFTWARE PROJECT #2: Does the software project have a community engagement / Q&A forum (self-hosted, on Stack Exchange etc.)?

<https://discourse.jupyter.org>

7. SOFTWARE PROJECT #2: Does the software project have contribution / coding guidelines?

<https://github.com/jupyter/notebook/blob/main/CONTRIBUTING.md>

8. SOFTWARE PROJECT #2: Is there a corresponding package available in a package manager (PyPi, CRAN, etc.)?

<https://pypi.org/project/notebook> <https://anaconda.org/conda-forge/notebook>

9. SOFTWARE PROJECT #2: Does the software project support continuous integration for testing?

GitHub workflows: <https://github.com/jupyter/notebook/tree/main/.github/workflows>

SOFTWARE PROJECT #2: Metrics- Impact (optional):

*Complete the following for the open source software project #2. **Providing metrics is optional and metrics can be approximate.** For each metric, please provide a source, clarify how the metric was computed, and/or provide any other comments. For monthly metrics, please provide data from the most recent month for which the corresponding metric is available.*

1. SOFTWARE PROJECT #2 : Complete the following table. List the number and explanation for each, if needed:

	Number	Comment
Scholarly paper(s) (including preprints) citing or mentioning the software project		
Monthly users, if applicable (based on one or more of the following: monthly downloads from websites, monthly downloads from package managers, monthly unique requests for updates, etc.)	11268923	Source: https://pepy.tech/project/notebook
Software projects that depend on the project		
Monthly visitors to project's website, discussion forum (e.g. Stack Overflow)		

2. SOFTWARE PROJECT #2: List of software projects to which key personnel on this proposal are contributing.

(No response)

SOFTWARE PROJECT #3

SOFTWARE PROJECT #3: Details:

Complete the following table for Open Source Software Project #3 of your proposal. All URLs should be in the format <https://example.com> and only one primary link should be provided.

Software Project name (required)	Lumino
Main code repository (e.g. GitHub URL) (required)	https://github.com/jupyterlab/lumino/
Homepage URL (required)	https://lumino.readthedocs.io/
Social media handles (if applicable)	(No response)
Do you or software project key personnel have commit rights to the code repositories for this software project? (required)	Yes
Short description of software project (200 words maximum) (required)	Lumino is a library for building interactive web applications.

SOFTWARE PROJECT #3: Metrics- Quality (required):

Complete for the open source software project #3.

1. SOFTWARE PROJECT #3 : What is the software project license?

Permissive license (e.g. BSD 3-Clause, MIT, Apache 2.0)

2. SOFTWARE PROJECT #3 : What is the main programming language?

Other (please specify): Typescript

3. SOFTWARE PROJECT #3: Does the software project have a code of conduct?

https://github.com/jupyter/governance/blob/master/conduct/code_of_conduct.md

4. SOFTWARE PROJECT #3: Does the software project have end-user documentation?

https://github.com/jupyterlab/.github/blob/master/CODE_OF_CONDUCT.md

5. SOFTWARE PROJECT #3 : Does the software project have an issue tracker?

<https://github.com/jupyterlab/lumino/issues>

6. SOFTWARE PROJECT #3: Does the software project have a community engagement / Q&A forum (self-hosted, on Stack Exchange etc.)?

<https://discourse.jupyter.org/>

7. SOFTWARE PROJECT #3: Does the software project have contribution / coding guidelines?

<https://github.com/jupyterlab/lumino/blob/main/CONTRIBUTING.md>

8. SOFTWARE PROJECT #3: Is there a corresponding package available in a package manager (PyPi, CRAN, etc.)?

<https://www.npmjs.com/package/@lumino/widgets>

9. SOFTWARE PROJECT #3: Does the software project support continuous integration for testing?

GitHub workflows: <https://github.com/jupyterlab/lumino/tree/main/.github/workflows>

SOFTWARE PROJECT #3: Metrics- Impact (optional):

*Complete the following for the open source software project #3. **Providing metrics is optional and metrics can be approximate.** For each metric, please provide a source, clarify how the metric was computed, and/or provide any other comments. For monthly metrics, please provide data from the most recent month for which the corresponding metric is available.*

1. SOFTWARE PROJECT #3 : Complete the following table. List the number and explanation for each, if needed:

	Number	Comment
Scholarly paper(s) (including preprints) citing or mentioning the software project		
Monthly users, if applicable (based on one or more of the following: monthly downloads from websites, monthly downloads from package managers, monthly unique requests for updates, etc.)	135755	Source: https://npm-stat.com/charts.html?package=%40lumino%2Fwidgets
Software projects that depend on the project	4181	Repositories depending on @lumino/widgets: https://github.com/jupyterlab/lumino/network/dependents?package_id=UGFja2FnZS03MDE4ODUzNTE%3D
Monthly visitors to project's website, discussion forum (e.g. Stack Overflow)		

2. SOFTWARE PROJECT #3: List of software projects to which key personnel on this proposal are contributing.

(No response)

SOFTWARE PROJECT #4

SOFTWARE PROJECT #4: Details:

Complete the following table for Open Source Software Project #4 of your proposal. All URLs should be in the format <https://example.com> and only one primary link should be provided.

Software Project name (required)	JupyterLab server
Main code repository (e.g. GitHub URL) (required)	https://github.com/jupyterlab/jupyterlab_server
Homepage URL (required)	https://jupyterlab-server.readthedocs.io/
Social media handles (if applicable)	(No response)
Do you or software project key personnel have commit rights to the code repositories for this software project? (required)	Yes
Short description of software project (200 words maximum) (required)	A set of server components for JupyterLab and JupyterLab like applications

SOFTWARE PROJECT #4: Metrics- Quality (required):

Complete for the open source software project #4.

1. SOFTWARE PROJECT #4 : What is the software project license?

Permissive license (e.g. BSD 3-Clause, MIT, Apache 2.0)

2. SOFTWARE PROJECT #4 : What is the main programming language?

Python

3. SOFTWARE PROJECT #4: Does the software project have a code of conduct?

https://github.com/jupyterlab/.github/blob/master/CODE_OF_CONDUCT.md

4. SOFTWARE PROJECT #4: Does the software project have end-user documentation?

<https://jupyterlab-server.readthedocs.io/>

5. SOFTWARE PROJECT #4 : Does the software project have an issue tracker?

https://github.com/jupyterlab/jupyterlab_server/issues

6. SOFTWARE PROJECT #4: Does the software project have a community engagement / Q&A forum (self-hosted, on Stack Exchange etc.)?

<https://discourse.jupyter.org/>

7. SOFTWARE PROJECT #4: Does the software project have contribution / coding guidelines?

https://github.com/jupyterlab/jupyterlab_server/blob/main/CONTRIBUTING.md

8. SOFTWARE PROJECT #4: Is there a corresponding package available in a package manager (PyPi, CRAN, etc.)?

<https://pypi.org/project/jupyterlab-server/> https://anaconda.org/conda-forge/jupyterlab_server

9. SOFTWARE PROJECT #4: Does the software project support continuous integration for testing?

GitHub workflows: https://github.com/jupyterlab/jupyterlab_server/tree/main/.github/workflows

SOFTWARE PROJECT #4: Metrics- Impact (optional):

*Complete the following for the open source software project #4. **Providing metrics is optional and metrics can be approximate.** For each metric, please provide a source, clarify how the metric was computed, and/or provide any other comments. For monthly metrics, please provide data from the most recent month for which the corresponding metric is available.*

1. SOFTWARE PROJECT #4 : Complete the following table. List the number and explanation for each, if needed:

	Number	Comment
Scholarly paper(s) (including preprints) citing or mentioning the software project		
Monthly users, if applicable (based on one or more of the following: monthly downloads from websites, monthly downloads from package managers, monthly unique requests for updates, etc.)	3317094	Source: https://pepy.tech/project/jupyterlab-server
Software projects that depend on the project		
Monthly visitors to project's website, discussion forum (e.g. Stack Overflow)		

2. SOFTWARE PROJECT #4: List of software projects to which key personnel on this proposal are contributing.

(No response)

SOFTWARE PROJECT #5

SOFTWARE PROJECT #5: Details:

Complete the following table for Open Source Software Project #5 of your proposal. All URLs should be in the format <https://example.com> and only one primary link should be provided.

Software Project name (required)	Jupyter server
Main code repository (e.g. GitHub URL) (required)	https://github.com/jupyter-server/jupyter_server/
Homepage URL (required)	https://jupyter-server.readthedocs.io/
Social media handles (if applicable)	(No response)
Do you or software project key personnel have commit rights to the code repositories for this software project? (required)	Yes
Short description of software project (200 words maximum) (required)	The backend—i.e. core services, APIs, and REST endpoints—to Jupyter web applications.

SOFTWARE PROJECT #5: List of Key Personnel:

Key personnel are people involved in the software project who will be supported by the grant if the application is successful.

Complete the following for the key personnel on the open source software project #5 (up to 5) (required); **enter n/a if any field is not applicable**. Personnel to be hired that have not been identified at this time can be listed in the budget section. You may need to use the scroll bar at the bottom of the table to scroll right to view and to complete all fields. Alternatively, you can tab to move through and complete the fields. **To add another person/row (up to five), click the box at the end of the row.**

	First name	Last name	Email address	Current employer /Affiliation	Job title	Developer username if applicable (e.g.,	Country of Residence	Add another person/row

3. SOFTWARE PROJECT #5: Does the software project have a code of conduct?

https://github.com/jupyter-server/.github/blob/main/CODE_OF_CONDUCT.md

4. SOFTWARE PROJECT #5: Does the software project have end-user documentation?

<https://jupyter-server.readthedocs.io/>

5. SOFTWARE PROJECT #5 : Does the software project have an issue tracker?

https://github.com/jupyter-server/jupyter_server/issues

6. SOFTWARE PROJECT #5: Does the software project have a community engagement / Q&A forum (self-hosted, on Stack Exchange etc.)?

<https://discourse.jupyter.org/>

7. SOFTWARE PROJECT #5: Does the software project have contribution / coding guidelines?

https://github.com/jupyter-server/jupyter_server/blob/main/CONTRIBUTING.rst

8. SOFTWARE PROJECT #5: Is there a corresponding package available in a package manager (PyPi, CRAN, etc.)?

<https://pypi.org/project/jupyter-server>

9. SOFTWARE PROJECT #5: Does the software project support continuous integration for testing?

GitHub workflows: https://github.com/jupyter-server/jupyter_server/tree/main/.github/workflows

SOFTWARE PROJECT #5: Metrics- Impact (optional):

Complete the following for the open source software project #5. **Providing metrics is optional and metrics can be approximate.** For each metric, please provide a source, clarify how the metric was computed, and/or provide any other comments. For monthly metrics, please provide data from the most recent month for which the corresponding metric is available.

1. SOFTWARE PROJECT #5 : Complete the following table. List the number and explanation for each, if needed:

	Number	Comment
Scholarly paper(s) (including preprints) citing or mentioning the software project		
Monthly users, if applicable (based on one or more of the following: monthly downloads from websites, monthly downloads from package managers, monthly unique requests for updates, etc.)	3352849	Source: https://pepy.tech/project/jupyter-server
Software projects that depend on the project		
Monthly visitors to project's website, discussion forum (e.g. Stack Overflow)		

2. SOFTWARE PROJECT #5: List of software projects to which key personnel on this proposal are contributing.

(No response)

8. Equal Opportunity & Diversity

Completed - Jun 2 2022

Equal Opportunity & Diversity

CZI Science supports the science and technology that will make it possible to cure, prevent, or manage all diseases by the end of this century. Everyone is affected by disease, yet different communities are affected by or experience disease in different ways. Moreover, due to systemic barriers, the scientific enterprise itself is not a place where all voices and talents thrive. We believe the strongest scientific teams — encompassing ourselves, our grantees, and our partners — incorporate a wide range of backgrounds, lived experiences, and perspectives that guide them to the most important unsolved problems. To enable our work, we incorporate diverse perspectives into our strategy and processes, and we also seek to empower community partners to engage in science.

We track demographic information associated with applications submitted to CZI in response to our open calls. This information helps us learn from the RFA process, as well as improve our strategies to help ensure members of underrepresented or marginalized groups in science are aware of and able to apply to CZI opportunities. **Please note that answering all questions below is voluntary, and demographic information will not be used to make final grant funding decisions.** All responses will be shared only with limited personnel, who will use that information only for the purposes described in this paragraph.

If you have any additional questions about why we ask this, what we do with the data, or to share suggestions for improvement, please reach out to sciencegrants@chanzuckerberg.com.

This section has been auto-filled based on data entered in the LOI for the Applicant. Please edit as needed. Please note that completing the below is voluntary, and demographic information will not be used to make final grant funding decisions.