JupyterIDE

Promoting JupyterLab features and extensions that facilitate collaboration among researchers and RSEs

Arizona State University

David Costello dcostel5@asu.edu
Nicole Brewer nbrewer6@asu.edu
Namita Shah nshah50@asu.edu

BACKGROUND

Research software engineers exist on a **continuum of different backgrounds** and skill sets, often using different tools and approaches for programming. Preferred development platforms each have their own advantages which cater to various users, but when programmers **collaborate closely** it can be beneficial to use a shared platform to encourage the shared use of shared practices. JupyterLab, an extensible platform, is capable of supporting a wide variety of development approaches and practices. We propose **JupyterLab as an ideal platform for collaboration** among programmers that prefer platforms ranging from Jupyter Notebooks to more traditional IDEs. We introduce JupyterIDE, a set of tutorials to assist interdisciplinary teams in converging on a set of shared best practices centered around JupyterLab.

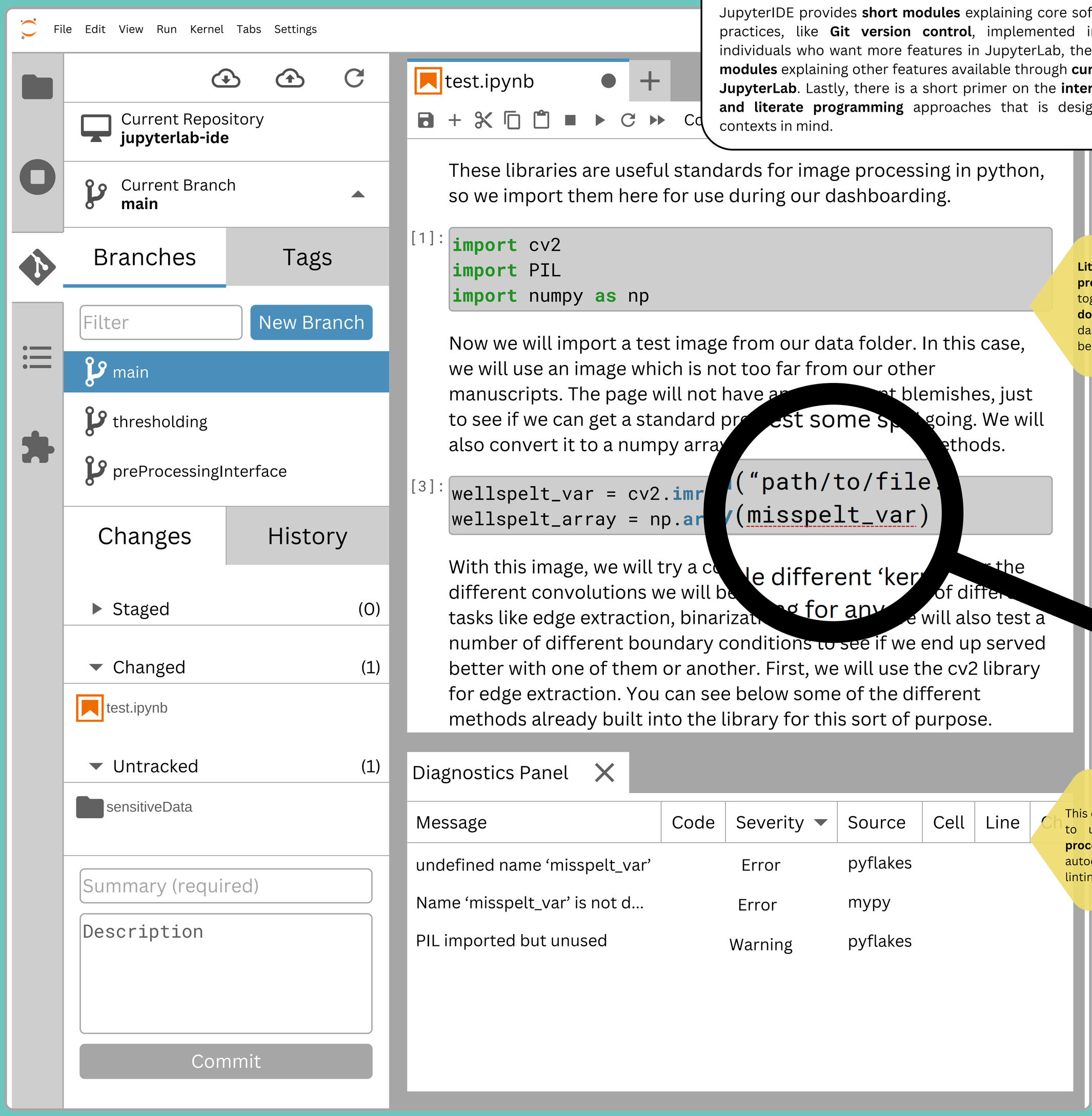
POTENTIAL NEEDS:

Domain Researchers and **Software Engineers** can benefit from a shared coding paradigm for research software

- Domain Researchers may benefit from a solid understanding of basic version control
- Software Engineers may want a development tool with more features

SOLUTION

JupyterIDE provides **short modules** explaining core software development practices, like **Git version control**, implemented in JupyterLab. For individuals who want more features in JupyterLab, there are **independent modules** explaining other features available through **curated extensions for JupyterLab**. Lastly, there is a short primer on the **intersection of iterative and literate programming** approaches that is designed with research contexts in mind.



Literate and iterative programming, used together, **create documentation** for dashboarding that can be replicated afterwards

jupyterlab-lsp

This **extension** allows users to use **language server processing**, including autocompletion, code linting, and **diagnostics**.

JUPYTER-IDE [ALSO] HAS:

Domain Researchers and **Software Engineers** can benefit from a short primer and example of literate programming for research contexts

- Domain Researchers may benefit from a self-paced tutorial on the Git extension and version control
- Software Engineers may want a curated list of extensions with tutorials giving users access to featurs like language server processing

REPO

Follow the link to access the **GitHub repository** and give it a

ACKNOWLEDGMENTS

better scientific software

This work was supported by the Better Scientific Software Fellowship Program, funded by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of the U.S. Department of Energy (DOE) Office of Science and the National Nuclear Security Administration; and by the National Science Foundation (NSF) under Grant No. 2154495.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DOE or NSF.