PyVO

Overview and community contribution refresher

Brigitta Sipőcz
Caltech/IPAC-IRSA

Active PyVO maintainers:
Adrian Damian (CADC), Markus Demleitner (Heidelberg University), Tom Donaldson (STScI)

Image by The Turing Way community and Scriberia used under a CC-BY licence.

Code on internet

≠

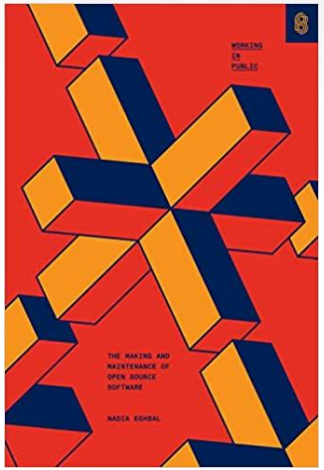
Open source

≠

Community development

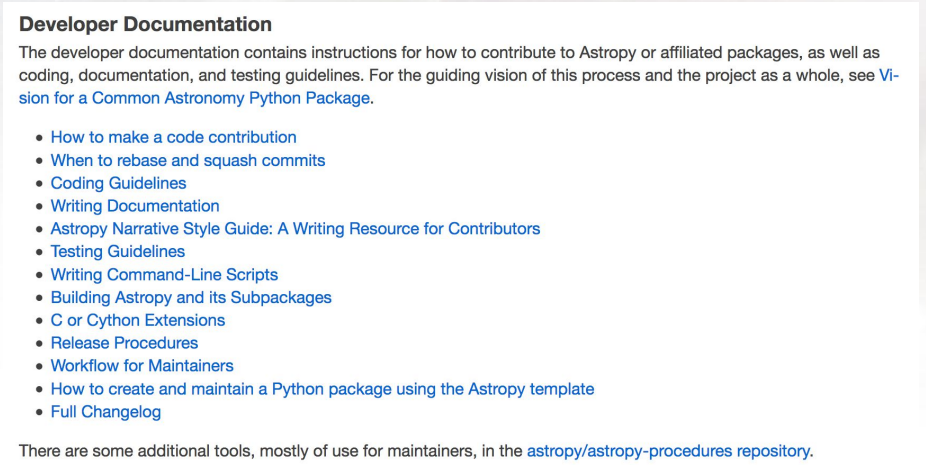
Collaborative software communities

- Community software
  - There is no central institute driving development
  - Open to and can handle community contributions
- A ***home platform***, accessible to anyone
  - GitHub, and services built around it
- Further reading on open-source communities
  - Nadia Eghbal/Asparouhova: Working in public



Open source infrastructure

- **Detailed developer guides**
  - Include developer tutorial, e.g.
    - how to report bugs (MWE)
    - a fully worked out PR example
- **Large number of checks on PRs**
  - Extensive CI testing
    - including testing with development versions of upstream dependencies, e.g. Python, Numpy, Matplotlib
  - Documentation build, and rendered version shared
  - Bots checking for codestyle, milestone, changelog etc



Developer Documentation

The developer documentation contains instructions for how to contribute to Astropy or affiliated packages, as well as coding, documentation, and testing guidelines. For the guiding vision of this process and the project as a whole, see Vision for a Common Astronomy Python Package.

- How to make a code contribution
- When to rebase and squash commits
- Coding Guidelines
- Writing Documentation
- Astropy Narrative Style Guide: A Writing Resource for Contributors
- Testing Guidelines
- Writing Command-Line Scripts
- Building Astropy and its Subpackages
- C or Cython Extensions
- Release Procedures
- Workflow for Maintainers
- How to create and maintain a Python package using the Astropy template
- Full Changelog

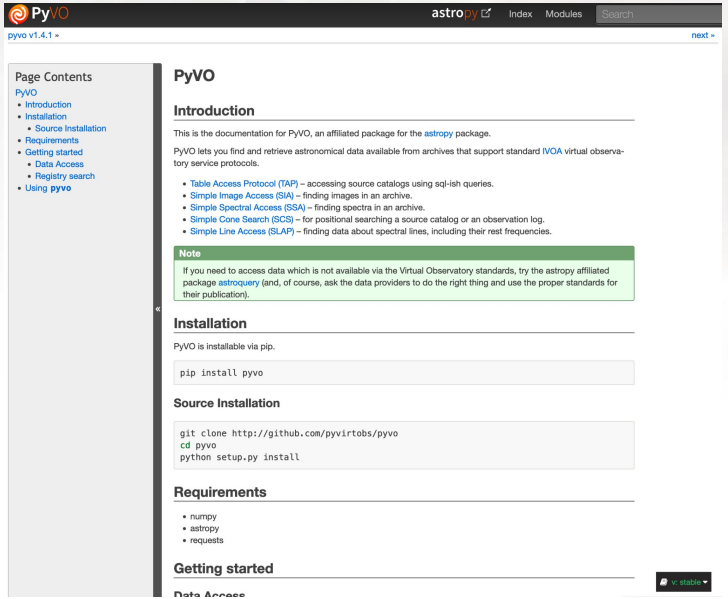
There are some additional tools, mostly of use for maintainers, in the astropy/astropy-procedures repository.

PyVO overview

Standalone Python library to support standard IVOA virtual observatory service protocols.

Docs:
https://pyvo.readthedocs.io/en/stable/

Code:
https://github.com/astropy/pyvo/



PyVO recent history

- Community restart at IVOA Interop 2019 Paris
    - Maintainer group to expand over multiple institutions
    - Tap into the infrastructure ecosystem of scientific python and astropy
- More flexibility on standards
    - prototype feature (see Omar Laurino's 2022 Apr Interop talk)

PyVO present

v1.5 (to be released soon)

- Registry search now finds SIA v2 services
- .to_qtable() method to return astropy QTable
- Make regtap service aware

- Cleanups and deprecations

- Various bugfixes since last interop, in v1.4.1

PyVO (near) future

v1.6 (~by next interop?)

- Empowered by prototype feature:
  - cloud access utilities (see Tess Jaffe's Science platform talk)

v...

- Possibilities that needs champions
  - Consolidating VO relevant pieces into PyVO

    (astropy.samp, and astroquery.vo_conesearch)
  - joining forces of **`astroquery.utils.tap`** (==TapPlus) and PyVO, possibly through prototype
  - Your feature

PyVO as backend

Direct usage is possible, but most astronomers will use it as a backend.

E.g. several astroquery modules:

- **`alma`**
- **`cadc`**
- **`ipac.nexsci`**
- More in the works: **`ipac.irsa`**, …

NAVO workshop tutorials