# PyVO

## Overview and testing refresher

PyVO maintainers:
Adrian Damian (CADC), Brigitta Sipőcz (Caltech/IPAC-IRSA), Manon Marchand (CDS),
Markus Demleitner (Heidelberg University), Tom Donaldson (STScI)

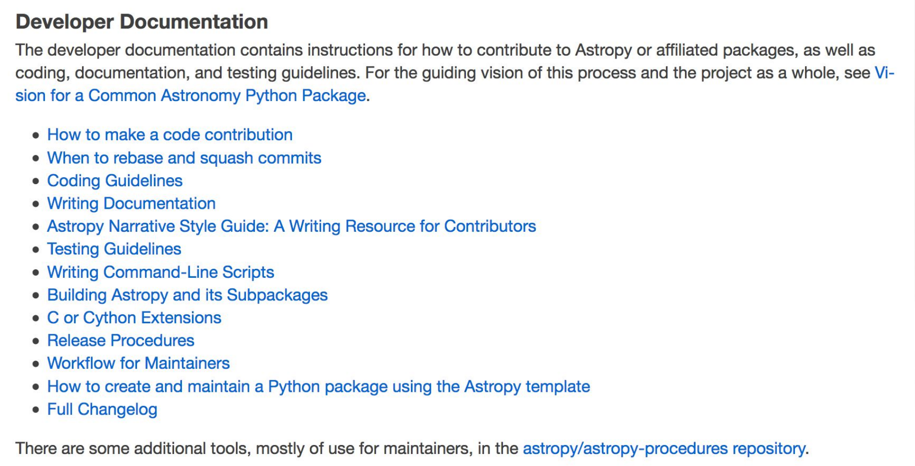
# Collaborative software communities

- 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

OPEN SOURCE COLLABORATION
BY THE COMMUNITY
FOR THE COMMUNITY
OPEN NOT SOURCE
DRIVEN BY NEEDS OF THE INSTITUTION OR THE INDIVIDUAL
Scriberia

# 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

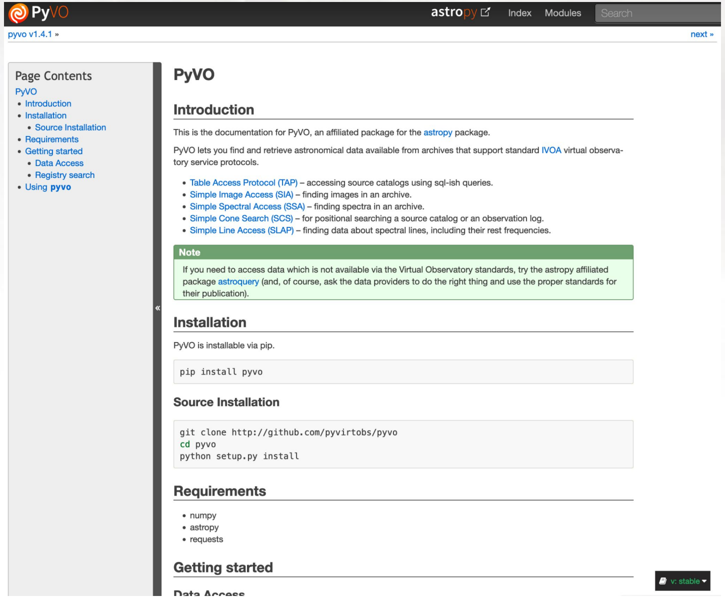
There are some additional tools, mostly of use for maintainers, in the astropy/astropy-procedures repository.

# PyVO overview

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

Docs:
[https://pyvo.readthedocs.io/en/stable/](https://pyvo.readthedocs.io/en/stable/)

Code:
[https://github.com/astropy/pyvo/](https://github.com/astropy/pyvo/)



Page Contents

## PyVO

### Introduction

This is the documentation for PyVO, an affiliated package for the astropy package.

PyVO lets you find and retrieve astronomical data available from archives that support standard IVOA virtual observatory service protocols.

- Table Access Protocol (TAP) – accessing source catalogs using sql-ish queries.
- Simple Image Access (SIA) – finding images in an archive.
- Simple Spectral Access (SSA) – finding spectra in an archive.
- Simple Cone Search (SCS) – for positional searching a source catalog or an observation log.
- Simple Line Access (SLAP) – finding data about spectral lines, including their rest frequencies.

**Note**

If you need to access data which is not available via the Virtual Observatory standards, try the astropy affiliated package astroquery (and, of course, ask the data providers to do the right thing and use the proper standards for their publication).

### Installation

PyVO is installable via pip.

```
pip install pyvo
```

#### Source Installation

```
git clone http://github.com/pyvirtobs/pyvo
cd pyvo
python setup.py install
```

### Requirements

- numpy
- astropy
- requests

### Getting started

#### Data Access

# PyVO recent history

- Maintainer group to expand over multiple institutions
  Newest member: **Manon Marchand** (CDS)

- Tap into the infrastructure ecosystem of scientific python and astropy

- More flexibility on standards
  - prototype feature available since 2022 Autumn and **in use** for
    - TAP1.2
    - MIVOT

# PyVO present

**v1.5.x** (Dec 2023-May 2024)

Overarching theme: ***make the package more generic***

- Make regTAP service aware
- Various SIA related fixes (registry search to find v2, fix not standard-mandated assumptions, etc)
- Various registry improvements
- Fixing bugs and compatibilities

**v1.6** (in development)

- MIVOT as prototype
- Global discovery as prototype (in PR)
- API cleanup, clarifications

# PyVO future

**V...**

- VOSpace client
- P3T
- download utilities
- 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 usage is indirect as a backend

E.g. several astroquery modules rely on TAP and/or SIA:

- **`alma, cadc, ipac.nexsci, ipac.irsa, simbad, vizier`**
- More to come: heasarc, simbad, …

Adding SSA usage to astroquery.ipac.irsa

# Importance of Downstream testing

Encourage testing of:

- Libraries including documentation, use-case examples and notebooks

Test as widely as possible for the supported use cases

- OS, supported dependency versions and development versions

Examples for PyVO downstream testing:

- astroquery test suite
- NAVO notebooks, IRSA notebooks, CDS notebooks
- Need more: e.g. ALMA notebooks, your notebooks?