PyVO Coordination

Nov 09, 2021

Attendees:

- Tom D - Developer at STScI/MAST, chair of IVOA Applications WG, one maintainer of astropy.io.votable and pyvo; interested in making pyvo as usable as possible for end users
- Adrian D - Vice-chair of Applications WG; writing python clients for CADC so wants to contribute to community efforts; contributor to ALMA module in astroquery
- Abdu Z - New scientist from HEASARC; hope to contribute to pyvo use cases and development
- Tess J - Science lead for NAVO; interested in making PyVO
- Brigitta S - Caltech/IPAC; maintainter of Astroquery; takes care of astroquery.ipac.irsa and overall maintanence
- Omar L - S/w eng. at Chandra X-Ray center; At CXO, use PyVO as client and have data holding that are published via VO so want to make sure they're accessible
- Renaud - Reasrch eng at Paris Observatory; use PyVO for several years now, esp. exploring registry, but also; have suite of tools for service validation
- Tamara C - Scientific Database and Web Access Engineer at CEFCA/OAJ; Interested in PyVO as it is the tool recommended to scientists; Having some issues with PyVO and want to help improve them
- Theresa D - Dev at STScI/MAST; improve client access and use
- Carlos B - Start using PyVO an VO back in 2015-16; Interest in PyVO, wrote a tool (https://github.com/chbrandt/eada) back in the day to harvest registry to look for conesearch services; wants to get back up to date on PyVO; maybe merge some tools back into PyVO
- Gilles L - CDS; Not a user of PyVO but interested in registry aspects esp.
- Doug B

A few possible points on the agenda:

- What should we do about experimental (non standard) code?
- Short term and long term priorities
- Feature requests (also ideas about how to engage contributors and the community in general)
- Governance and contribution guidelines
- Table upload feature
- Just a note that there are pending infrastructure improvements from Astropy

What should we do about experimental (non standard) code?

Can be either something experiemental or code that is implementing candiate standards.

Current code in PyVO follows specs, so works with services that follows the specs.

Brigitta.: From user perspective, if these features could be shipped to user as part of the package (no need for separate branch) that is more useful. E.g., TAPPlus should be in PyVO proper whether or not there are pending standard issues.

Carlos: Where can I find these features?
Adrian: Not currently in PyVO, but are considering adding table upload for TAP. Would be proposed for TAP 1.2, but not currently clearly supported by standards. Would work with CADC, but not other TAP services, depending on how the standard goes. Something like the Astroquery model, where there is some client code that supports server-specific features.
Brigitta: Not suggesting cadctap module in PyVO, but make it more generic. Use documentation, possible warnings to clarify the experiemental feature.
Adrian: There might be competing implementations
Brigitta: Might help force collaboration.
Tess: Is this upload for cross-correlation?
Adrian: No, this is not for temporary table to do a query; it's for a more permanent table, but not supported by standard
Tess: Error trapping in PyVO needs improvement and this may be an example. How best to tell users that "this service does not support that"? This is an argument towards having these features in PyVO. We already have sort of mixed compatibility/interoperability and should be giving useful feedback to users
Tom: Having these developments in full public view to be exercised could be a very healthy way to evolve potential standards, naturally moving from idea to convention to standard, all with potential for community participation and input.
Carlos: From eng. point of view a plug-in type of mechanism might be a cleaner way keep these features separate.
Omar: Experience with feature flags? Not done it myself, but hear it is a fahionable way to support this sort of keeping these features just separate enough.
Carlos: With decorators, would bring warning.
Adrian: Let's move forward with this table upload feature PR:
https://github.com/astropy/pyvo/pull/274

Short term and long term priorities
Adrian: Like to see higher-level features. Envision application(s) which hide details (and lingo) of standards that one could show to astronomers.
Tom: NAVO also interested in that. Slowly working in that direction via workshop science use case notebooks.

Tess: NAVO has been collecting list of bugs and features they would like to improve:

- SIA:size, SSA:diameter, SCS:radius, astroquery:radius|width|height. Common pyvo option that converts as needed? [hard]
- Expose TAPService.tables in RegistryResource [easy?]
- Expose TAPService.examples in RegistryResource [easy?]
- Table column describe() method [easy?]
- Method in pyvo.io.vosi.vodataservice.Table to get columns corresponding to a given UCD [easy?]
- Sensible return from getdatalink() if there is no datalink [easy]
- Method to get the contact info email for the curator responsible for a given service [?]
- Insert hooks to make raw VOTable results available to user.

- Improve TAP error handling when services return error info in different ways.
- Fix async with catalog upload (or is this heasarc? it works sync)

Brigitta: Recommend filing issues. People can comment/clarify/fix/close... Sometimes better to have issue prior to implementing a fix. Better chance for consensus.

PyVO repo, with issues, PRs, etc. is https://github.com/astropy/pyvo

Adrian: If someone can specify from users' perspective what I need to get my job done, then that's approaching the standard from the top down.

Abdu: Those user-centric perspectives will be part of my job moving forward.

Tamara: Possibility to indicate the different formats in results. E.g., can we get results in csv, fits, etc. (binary to improve performance, but also users are just more ready to use certain formats). (https://github.com/astropy/pyvo/issues/149)

Gilles: To search data in registry. This is possible today via keywords. Can we add search by position for example, esp. when resources have associated MOCs. Maybe by author as well.

Adrian: Any suggestions on other ways to engage contributors?

Brigitta: May be harder with PyVO than Astroquery because in the latter you just implement what user wants. The framework is less restrictive. In general, just keep repeating encouragement to contribute issues, comments, PRs via mail lists, Slack and github.

Adrian: I think we can do better on documentation. There are efforts on tutorials, etc., but we can link to those and maybe include more robust examples on the top level docs.

Tom: Work expected soon on VODML metadata in VOTable

Omar: Going back to experimental features, would it be useful for me to pursue that?
All: Yes!

Tom: Do this again? How often? Every couple months? OK sometime in Jan or Feb.

Reminder to participate at Astropy#pyvo and IVOA#applications Slack channels as well as pyvo repo: https://github.com/astropy/pyvo

Post-meeting note:

An example of how I'd implement those decorators: https://github.com/docdatpie/experimental_warnings_layout