DAL Splinter - Interop. Nov 2025

Date: 15 Nov. 2025 - 17:00

Attendees: Pat Dowler, Paul Harrison, Jesus Salgado, Joshua Fraustro, Marco Molinaro, Grégory Mantelet

Goal: to agree on a sequence (ideally a timeline) to update TAP, UWS and VOSI in order to have OpenAPI available in all these standards.

Useful links

- TAP : https://github.com/pdowler/ivoa-TAP/tree/main/openapi
- VOSI: https://github.com/ivoa-std/VOSI/tree/main/openapi

Discussion

- Paul wants to try a new implementation of UWS based on old astrogrid code.
  - A new version of UWS may be based on the implementations experiences
  - UWS document is on GitHub (https://github.com/ivoa-std/UWS)), but there is no .tex file and no ivoatex ; the source document is HTML.
- TAP-1.2, mostly with user table management
  - OpenAPI defined in VOSI (vosi-table.yml) and theoretically in UWS (partial OpenAPI documents are in the TAP's GitHub repo)
  - Implemented in YouCat (CADC) et DaCHS (GAVO)
- Links between OpenAPI pieces spread over VOSI, DALI, UWS and TAP standards made using $ref (can be local files or URL)
- TAP and UWS interleave (async endpoint)
  - kind of duplication of all UWS endpoints in TAP's OpenAPI ; for the moment only /async/JOB-ID and /async/JOB-ID/phase should be enough.
  - OpenAPI does not allow us (for now) to extend a description by adding new parameters
  - We could add two more lines for the result, although parsing the job description is enough to find the result, error or anything else:
    - /results
    - /results/result , which is TAP specific (does not apply to VOSpace)
- Pat and Joshua have a minimal OpenAPI description of UWS
  - Pat volunteered to complete this OpenAPI document for UWS.
- Marco: if minor change, Errata + OpenAPI, but if major change a new version is needed

Some meta questions/comments (Paul Harrison)

- is there an agreed tool that is going to read the OpenAPI specs and produce products from it - at a minimum documentation, and better still client code.?
    - Pat,Joshua: No specific agreed tool but there are tools (mostly command line) to validate an OpenAPI document. VSCode is also used to quickly edit and see errors.
- Such a tool needs to be included into the GitHub CI so that we can validate any specs produced.
    - *Not answered*
- It is potentially difficult to get the OpenAPI tooling that I have used to pick up the modular structure that would be best - especially if it is not in the same repository - I think that Dave Morris has some more experience in this..
    - *Not answered*

Conclusion/Plan

1. Finish writing the OpenAPI for UWS-1.1
2. If minor change needed, publish Errata for UWS-1.1. If major change, a new version of UWS will be needed.
3. Publish the OpenAPI on IVOA doc repo along with the UWS-1.1 doc.