NRC·CNRC

WD-TAP-1.2 Status

Patrick Dowler
Canadian Astronomy Data Centre

DAL session 1 - 2026-06-09

Overview

- assembly & validation
- define TAP using OpenAPI
- re-use OpenAPI components from other standards: VOSI, DALI, UWS
- re-use existing file formats (primarily VOTable)
- ~compatible with TAP-1.1

OpenAPI: assembling a complete standard

- IVOA standards are based on other more foundational standards
- they are already assembled (by citation)
- modularity: $ref mechanism
- assembly:
  - import step and then $ref to a local copy
  - $ref directly to the component using a URL
  - import: automated or explicit standard developer action?

OpenAPI: TAP standard

```
paths:
  /capabilities:
       $ref: ./openapi/vosi/vosi-capabilities.yaml
  /tables:
    $ref: ./openapi/vosi/vosi-tableset.yaml
  /tables/{name}:
    $ref: ./openapi/vosi/vosi-table.yaml

  /table-ops:
    $ref: ./openapi/vosi/vosi-table-ops.yaml
  /table-ops/{jobID}:
    $ref: ./openapi/vosi/vosi-table-ops-job.yaml
  /table-ops/{jobID}/phase:
    $ref: ./openapi/vosi/vosi-table-ops-job-phase.yaml
```

```
/sync:
   $ref: ./openapi/tap/tap-sync.yaml

/async:
   $ref: ./openapi/tap/tap-async.yaml
/async/{jobID}:
   $ref: ./openapi/tap/tap-async-job.yaml
/async/{jobID}/phase:
   $ref: ./openapi/tap/tap-async-job-phase.yaml
/async/{jobID}/results/result:
   $ref: ./openapi/tap/tap-result.yaml
```

OpenAPI: assembling a complete standard

```
paths:
   /capabilities:
       $ref: https://www.ivoa.net/std/openapi/vosi/{major?}/vosi-capabilities.yaml

   /tables:
      $ref: https://www.ivoa.net/std/openapi/vosi/{major?}/vosi/vosi-tableset.yaml
   /tables/{name}:
      $ref: https://www.ivoa.net/std/openapi/vosi/{major?}/vosi/vosi-table.yaml

etc…
```

OpenAPI: validation

- using two clients to check that the assembled OpenAPI is right
  - https://github.com/daveshanley/vacuum OpenAPI 3.1 validation
  - Swagger UI to render API docs in web browser, eg: https://ws-uv.canfar.net/youcat/
- for validation:
  - no errors, no warnings
  - info for each use of $ref acceptable
  - current PR https://github.com/ivoa-std/TAP/pull/8

TAP-1.2 OpenAPI - complications

- two UWS endpoints: vosi-table-ops & tap-async
  - same GET parameters (from UWS)
  - different POST parameters
  - unique operationId
  - therefore: two sets of yaml files using UWS pattern
- two ways to signal optional features:
  - for optional HTTP methods in VOSI tables: 405
  - for optional HTTP endpoint (eg vosi-table-ops, examples): 404
- misc
  - OpenAPI response model implies use of accepts header
  - DALI responseformat controls output format for TAP-sync and -async
  - optional formats??

OpenAPI for TAP-1.2 – UWS in tap-async.yaml

```
get:
   operationId: query-job-list
   tags:
     - TAP Query
   summary: list queries (UWS jobs)
   description: TAP asynchronous query endpoint
   parameters:
     - $ref: ../uws/uws-phase-param.yaml
     - $ref: ../uws/uws-after-param.yaml
     - $ref: ../uws/uws-last-param.yaml
   responses:
     '200':
       $ref: ../uws/uws-responses.yaml#/job-listing

etc…
```

```
post:
   operationId: query-job-create
   tags:
     - TAP Query
   summary: create async query (UWS job)
   description: TAP asynchronous query endpoint
   parameters:
     - $ref: ./tap-lang.yaml
     - $ref: ./tap-query.yaml
     - $ref: ./tap-format.yaml
     - $ref: ../dali/dali-maxrec.yaml
     - $ref: ../dali/dali-responseformat.yaml
     - $ref: ../dali/dali-upload.yaml
   responses:
     '303':
       $ref: ../uws/uws-responses.yaml#/created
```

OpenAPI for TAP-1.2 - complications

- case sensitivity of parameter names
  - OpenAPI has 3 types of parameters: header, query, path
    - headers are not case sensitive, recent s/w -> lower
    - query param names are case sensitive
    - path param names are case sensitive
  - **** DALI-1.1 says query param names are not case sensitive ****

  - what to do and when?
  - thought to postpone this, but RFC comments suggest we should address it in PR-DALI-1.2

questions?