

TAP-next: User-managed Tables in TAP

Patrick Dowler
Canadian Astronomy Data Centre

people: Brian Major, Adrian Damian



Topics

- API extensions for TAP: implemented at CADC and in use by projects (mainly CANAFR youcat service)
- open questions/concerns
- API extensions: requested/ideas - not implemented

Context

- CANAFR youcat service: <ivo://cadc.nrc.ca/youcat>
 - schemas: 5
 - tables: 81
 - columns: 5210
 - primarily astronomical source catalogues
- permissions
 - 4 public, 1 private schema
 - 28 public tables: anon query
 - 53 private tables: 40 owner only, 13 readable (queryable) by group
 - 18 tables where a group has been granted read-write (insert)
- metadata for those 5210 columns
 - 4491 real: 3216 float, 1242 double, 29 point, 4 polygon
 - 5% of “real” columns have a unit
 - 29% of columns have a description
 - 3% of columns have UCDs

Context: existing software

- server: <https://github.com/opencadc/tap>
 - most code implemented in the libraries
 - complete youcat service build available
 - prebuilt docker image: images.opencadc.org/core/youcat
 -
 - limitations: currently PostgreSQL+pgsphere only
 - important admin not documented
 - several critical TODOs documented
 - future plans documented
- client: cadctap client
- client: pyvo feature branch (Adrian Damian)

Necessary API

- list tables: VOSI-tables (or tap_schema)
- get table metadata: VOSI-tables (or tap_schema)
 - **GET /youcat/tables/{table_name}**
- create table
 - **PUT /youcat/tables/{table_name}**
- update table metadata
 - **POST /youcat/tables/{table_name}**
- drop table
 - **DELETE /youcat/tables/{table_name}**

Necessary API

- load data
 - **POST /youcat/load/{table_name}**
 - this designed mainly around good clean error reporting and recovery
 - it is really “append rows” so client can gradually add data and can resume after partial success
- create index (async: parameter-style UWS job)
 - **POST /youcat/table-update**
 - TABLE={table_name}
 - INDEX={column_name}
 - UNIQUE=true (default: false)
 - param limitations: single column only
 - why async? can be done after table load so it takes time, service can control/schedule execution of such jobs
 - checks and updates tap_schema.columns.indexed

Necessary API

- get permissions
 - **GET /youcat/permissions/{name}**
 - name: schema or table name
 - document: simple ascii key=value
- set permissions
 - **POST {document} /youcat/permissions/{name}**
 - doc specifies all permissions: single call, also removes
 - property names, cardinality, file format all rough prototype

Permissions: public vs project/team vs private

- schema and table properties:
 - owner
 - anonymous read
 - group read permission
 - group with read-write permission
- on schema: read permission lets you list tables
 - effects: VOSI-tables and query on tap_schema
- on schema: write permission lets you create and drop tables
- on table: read permission lets you query
 - effects: query validation, not found vs permission denied
- on table: write permission lets you modify metadata, append rows, create index
- owner: read-write + change permissions + drop table

Open questions

- prototype involves new endpoints
 - TAP standardID fragments?
 - registry extension with <endpoints>...
- prototype extends VOSI-tables (more HTTP verbs)
 - registry extension to list supported verbs
 - modify VOSI?
 - define extension in TAP?
 - create a new standard that defines the extension?

Open questions

- publish table metadata to IVOA registry?
 - table metadata under user control: dynamic
 - changes have to propagate: temporary discrepancies (reality)
 - opt-in: publish has to be chosen/enabled by owner (opinion)
 - explicit publish action by owner? part of API or site-specific?
 - conservative: only publish public (anon queryable) tables? at least to start...
 - service operator responsible for metadata “quality” review?

Other API

- augment tap_schema with new schema and table properties?
 - enables users to manage their content
 - exposes permissions to users of TAP so they can understand some failures and ask for permission
 - youcat: implemented in tap_schema but not exposed
 - would require 4 new optional columns:
 - owner datatype="char" arraysize="*"
 - anon_read datatype="boolean" (or "int"?)
 - group_read datatype="char" arraysize="*" xtype="uri"
 - group_write datatype="char" arraysize="*" xtype="uri"

Other API aka “things people asked for”

- rename table or column?
 - dynamic table metadata
- multi-column index??
 - not describable in tap_schema right now
- explicit primary key?
 - not explicit in tap_schema right now
- declare foreign keys??
 - add to tap_schema.keys, also modify? remove?
 - constraint optional? can just be a “how to join”
- update: replace rows by PK
 - requires explicit PK above
 - explicit update or just **re-load**?
 - single row? size limit? (intent: one transaction so modest size)
- delete rows with WHERE clause?
 - async job
 - probably has to be batched into multiple transactions like load

Summary

- extend TAP standard to support user-managed tables: TAP-1.2
 - all features would be optional
 - extend VOSI-tables API
 - endpoints for uploading rows, permissions, async jobs
 - extend tap_schema to permission permissions
- publish user tables to IVOA registry
- wishlist of additional features: less necessary and harder to do

shameless plug

<https://github.com/opencadc/tap>