Metadata Templates
Metadata templates enable users and curators to:

- View and interact with AVUs in a *user-friendly* interface
- *Require* metadata attributes on iRODS objects
- *Validate* metadata elements
- *Standardize* metadata associated with iRODS objects
- *Uniformly apply* metadata to many objects simultaneously
- Provide *rendering guidance* to GUIs
Available in white paper
  • not reproduced here for space/readability

A Metadata Template contains the following properties:
  • name
  • type
  • source
  • destination
  • description
  • author
  • version
  • required
  • elements
Metadata Template type

An enum indicating what type of Metadata Template this is

Currently, only FORM_BASED Metadata Templates are supported

We have discussed adding, for example, support for Templates derived from schema.org schema (SCHEMA_REF)
An enum indicating where the data to populate the Metadata Template will come from

Currently, the only supported source is USER

We have discussed adding, for example, the ability to populate a Metadata Template with output from an iRODS rule (RULE) or a combination of rule output and user input (MIXED)
An enum indicating how the metadata will actually be stored on disk

Currently, the only supported destination is IRODS, indicating that the metadata will be stored as AVU triples in the iRODS catalog

We have discussed adding, for example, the ability to store metadata in an external Postgres database (POSTGRES), as is done in CyVerse
Available in white paper
  • not reproduced here for space/readability

A Metadata Template contains the following properties:
  • name
  • i18nName
  • description
  • i18nDescription
  • type
  • source
  • defaultValue
  • validationStyle
  • validationOptions
  • required
An enum indicating how/if a metadata element will be validated

Supported validation styles are the following:

- DEFAULT
- IS
- IN_LIST
- IN_RANGE
- IN_RANGE_EXCLUSIVE
- REGEX
- FOLLOW_REF
- DO_NOT_VALIDATE
An enum indicating how/if a metadata element will be validated

Supported validation styles are the following:

- DEFAULT
- IS
- IN_LIST
- IN_RANGE
- IN_RANGE_EXCLUSIVE
- REGEX
- FOLLOW_REF
- DO_NOT_VALIDATE
Supported Metadata Element types

- RAW_STRING
- RAW_TEXT
- RAW_URL
- RAW_INT
- RAW_FLOAT
- RAW_BOOLEAN
- RAW_DATE
- RAW_TIME
- RAW_DATETIME
- REF_IRODS_QUERY
- REF_IRODS_CATALOG
- REF_URL
- LIST_STRING
- LIST_INT
- LIST_FLOAT
Components of the Metadata Template architecture

Parser
- Generates MetadataTemplate POJOs from JSON and vice versa

Validator
- Validates Metadata Templates and Elements

Resolver
- Handles find/list/CRUD operations on template files

Exporter
- Saves populated Metadata Templates to permanent metadata store
Metadata Templates + FormBot in use

User
- User uploads file
- Populates HTML forms with required/desired values, attempts to save

Front End
- Checks to see if there are any required templates by calling listRequiredTemplates
- Renders list of HTML forms
- For each form, calls executeFormBotForm
- Displays result of execution

FormBot
- For each form, generates a populated Metadata Template
- Validates each MetadataTemplate

MetadataTemplates
- Builds a list of JSON representations of HTML forms for each MetadataTemplate
- Builds a list of MetadataTemplates from required .mdtemplate files found in .irods collections

iRODS (Jargon)
- Opens .mdtemplate files
- Executes each MetadataTemplate
- Writes AVUs onto file(s)

Returns