

# Python package for metadata schemas

Mariana Montes and Ronny Moreas

2024-05-29

# Outline

- The ManGO Metadata Schema Manager
- From JSON to validation
- From a Python dictionary to AVUs
- Conclusion

# The ManGO Metadata Schema Manager

# Form to add metadata

**datateam\_icts\_icts\_test**

---

Collections

Search

Metadata schemas

Group administration

Trash

---

Admin

## detal&doe2024.pdf

Type: data\_object

Realm: datateam\_icts\_icts\_test

### Metadata schema: Book schema as an example 2.0.0

#### Book title\*

Input type: text

#### Publishing date\*



Input type: date

#### Colors in the cover

- red
- blue
- green
- yellow

#### Publishing house\*

# View of the schema metadata

The screenshot displays the iRODS web interface. At the top left, the logos for KU LEUVEN, ManGO, and icts are visible. A search bar is located at the top center with the placeholder text "Search file names, collection names, owner names and selected metadata fields...". On the right side of the top bar, there is a user profile icon and a dropdown arrow.

The main content area is divided into a left sidebar and a main panel. The sidebar contains the following items:



- datateam\_icts\_icts\_test (with a red share icon)
- Collections
- Search
- Metadata schemas
- Group administration
- Trash
- Admin (with a wrench icon)

The main panel shows the breadcrumb path: [icts](#) » [home](#) » [datateam icts icts test](#) » [irodsugm\\_demo](#). Below this is the file name **deta&doe2024.pdf** with a download icon to its right. A tabbed interface is present with the following tabs: System properties, **Metadata**, Permissions, Preview, and Metadata inspection and extraction. Under the Metadata tab, there is a button labeled "Book schema as an example" and the text "Schema version: 2.0.0".


The metadata details are as follows:


- Book title**: An exemplary book
- Publishing date**: 2024-02-01
- Colors in the cover**: red, yellow
- Publishing house**: Tor


# The Schema Manager


 **datateam\_icts\_icts\_test** 


---

 Collections


 Search


 Metadata schemas

 Group administration

 Trash

---


Admin 

Book schema as an example 

version 2.0.0 published

[View](#) [New \(draft\) version](#) [Copy to new schema](#) [Download JSON](#) [Archive](#)

**Book title\***  
Input type: text

**Publishing date\***   
Input type: date

**Colors in the cover**

red  
 blue  
 green  
 yellow

**Publishing house\***

**Is there an e-book**

Available  
 Unavailable

# Metadata Schemas as JSON

## book-v2.0.0-published.json

```
{'schema_name': 'book',  
  'version': '2.0.0',  
  'status': 'published',  
  'properties': {'title': {'title': 'Book title',  
    'type': 'text',  
    'required': True},  
    'publishing_date': {'title': 'Publishing date',  
      'type': 'date',  
      'required': True,  
      'repeatable': True},  
    'cover_colors': {'title': 'Colors in the cover',  
      'type': 'select',  
      'values': ['red', 'blue', 'green', 'yellow'],  
      'multiple': True,  
      'ui': 'checkbox'}},  
  'publishing_date': {'title': 'Publishing date'}}
```

# Minimal example

```
1 pip install mango-mdschema
```

## add\_schema\_metadata.py

```
1 import json
2 from irods.session import iRODSSession
3 from mango_mdschema import Schema
4
5 with open("metadata_file.json", "r") as f:
6     my_metadata = json.load(f) # a dictionary
7
8 my_schema = Schema("book-v2.0.0-published.json")
9 with iRODSSession(irods_env_file=env_file) as session:
10     irods_object = session.collections.get(home_dir).data_objects[0]
11     my_schema.apply(irods_object, my_metadata) # includes validation
```



# From JSON to validation

# Metadata schemas as JSON

## book-v2.0.0-published.json

```
{'schema_name': 'book',  
  'version': '2.0.0',  
  'status': 'published',  
  'properties': {'title': {'title': 'Book title',  
    'type': 'text',  
    'required': True},  
    'publishing_date': {'title': 'Publishing date',  
      'type': 'date',  
      'required': True,  
      'repeatable': True},  
    'cover_colors': {'title': 'Colors in the cover',  
      'type': 'select',  
      'values': ['red', 'blue', 'green', 'yellow'],  
      'multiple': True,  
      'ui': 'checkbox'}},  
  'publishing_date': {'title': 'Publishing date'}}
```

# Interpretation via the Python package

```
1 book_schema = Schema("book-v2.0.0-published.json")  
2 print(book_schema)
```

①

Book schema as an example

Metadata annotated with the schema 'book' (2.0.0) carry the prefix 'mgs'.

This schema contains the following 7 fields:

- title, of type 'text' (required).
- publishing\_date, of type 'date' (required).
- cover\_colors, of type 'select'.
- publisher, of type 'select' (required).
- ebook, of type 'select'.
- author, of type 'object' (required).
- market\_price, of type 'float'.

# Field requirements

```
1 book_schema.print_requirements("publishing_date")
```

Type: date.

Required: True. Default: None.

Repeatable: True.

```
1 book_schema.print_requirements("publisher")
```

Type: select.

Required: True. Default: Tor.

Repeatable: False.

Choose only one of the following values:

- Penguin House
- Tor
- Corgi
- Nightshade books

# Field requirements

```
1 book_schema.print_requirements("cover_colors")
```

Type: select.

Required: False.

Repeatable: False.

Choose at least one of the following values:

- red
- blue
- green
- yellow

# From a Python dictionary to AVUs

# Required fields and defaults

```
1 my_metadata = {  
2     "title": "An exemplary book",  
3     "author": [  
4         {"name": "Fulano De Tal", "email": "fulano.detal@kuleuven.be"},  
5         {"name": "Jane Doe", "email": "jane.doe@kuleuven.be"},  
6     ],  
7     "ebook": "Available",  
8     "publishing_date": "2024-02-01",  
9     "cover_colors": ["red", "magenta", "yellow", "turquoise"],  
10 }  
11 book_schema.validate(my_metadata)
```

```
{'title': 'An exemplary book',  
'author': [{'name': 'Fulano De Tal', 'email': ['fulano.detal@kuleuven.be']},  
            {'name': 'Jane Doe', 'email': ['jane.doe@kuleuven.be']}],  
'ebook': 'Available',  
'publishing_date': [datetime.date(2024, 2, 1)],  
'cover_colors': ['red', 'yellow'],  
'publisher': 'Tor'}
```

# Error messages

```
1 book_schema.validate(  
2     {  
3         "title": "Some title",  
4         "author": {"name": "Jane Doe", "email": "sweetdoe@email.eu"},  
5         "publishing_date": date.today(),  
6     }  
7 )
```

ValidationError: 'book.author.email' does not match pattern  
'^[^@]+@kuleuven.be\$', got value 'sweetdoe@email.eu'

```
1 book_schema.validate(  
2     {  
3         "title": "Some title",  
4         "author": {"name": "Jane Doe", "email": "jane.doe@kuleuven."},  
5         "publishing_date": "01/01/1990",  
6     }  
7 )
```

ConversionError: 'book.publishing\_date' cannot be converted to a date, got  
value '01/01/1990'



# Warnings

```
1 import logging
2
3 logger = logging.getLogger("mango_mdschema")
4 logger.setLevel(logging.INFO)
5
6 book_schema.validate(my_metadata)
```

```
INFO:mango_mdschema:Applying default value to required field 'book.publisher':
'Tor'
```

```
INFO:mango_mdschema:Some values in 'book.cover_colors' were not allowed and
are discarded: magenta, turquoise. Allowed values: red, blue, green, yellow.
```

```
INFO:mango_mdschema:Missing non-required fields in 'book': ['market_price']
```

```
INFO:mango_mdschema:Missing non-required fields in 'book.author': ['age']
```

```
INFO:mango_mdschema:Missing non-required fields in 'book.author': ['age']
```

```
{'title': 'An exemplary book',
 'author': [{'name': 'Fulano De Tal', 'email': ['fulano.detal@kuleuven.be']},
            {'name': 'Jane Doe', 'email': ['jane.doe@kuleuven.be']}],
 'ebook': 'Available',
 'publishing_date': [datetime.date(2024, 2, 1)],
 'cover_colors': ['red', 'yellow'],
 'publisher': 'Tor'}
```

# Write: from dictionaries to namespaces

```
1 irods_object = session.collections.get(home_dir).data_objects[0]
2 irods_object.metadata.items()
```

```
[]
```

```
1 avus = book_schema.to_avus(my_metadata)
2 avus
```

```
[<iRODSMeta None mgs.book.title An exemplary book None>,
 <iRODSMeta None mgs.book.author.name Fulano De Tal 1>,
 <iRODSMeta None mgs.book.author.email fulano.detal@kuleuven.be 1>,
 <iRODSMeta None mgs.book.author.name Jane Doe 2>,
 <iRODSMeta None mgs.book.author.email jane.doe@kuleuven.be 2>,
 <iRODSMeta None mgs.book.ebook Available None>,
 <iRODSMeta None mgs.book.publishing_date 2024-02-01 None>,
 <iRODSMeta None mgs.book.cover_colors red None>,
 <iRODSMeta None mgs.book.cover_colors yellow None>,
 <iRODSMeta None mgs.book.publisher Tor None>]
```

# Write: from dictionaries to namespaces

```
1 book_schema.apply(irods_object, my_metadata)
2 irods_object.metadata.items()
```

```
[<iRODSMeta 8475276 mgs.book.cover_colors red None>,
 <iRODSMeta 8497207 mgs.book.title An exemplary book None>,
 <iRODSMeta 8497210 mgs.book.author.name Fulano De Tal 1>,
 <iRODSMeta 8497213 mgs.book.author.email fulano.detal@kuleuven.be 1>,
 <iRODSMeta 8497216 mgs.book.author.name Jane Doe 2>,
 <iRODSMeta 8497219 mgs.book.author.email jane.doe@kuleuven.be 2>,
 <iRODSMeta 8497222 mgs.book.ebook Available None>,
 <iRODSMeta 8497225 mgs.book.publishing_date 2024-02-01 None>,
 <iRODSMeta 8497228 mgs.book.cover_colors yellow None>,
 <iRODSMeta 8497231 mgs.book.publisher Tor None>,
 <iRODSMeta 8497234 mgs.book.__version__ 2.0.0 None>]
```

# Read: from AVUs back to dictionaries

```
1 #book_schema.from_avus(avus)
2 book_schema.extract(irods_object)
```

```
{'author': [{'email': ['fulano.detal@kuleuven.be'], 'name': 'Fulano De Tal'},
             {'email': ['jane.doe@kuleuven.be'], 'name': 'Jane Doe'}],
 'cover_colors': ['red', 'yellow'],
 'ebook': 'Available',
 'publisher': 'Tor',
 'publishing_date': [datetime.date(2024, 2, 1)],
 'title': 'An exemplary book'}
```

# Conclusion

# Metadata schemas with Python

## Metadata schemas

- Format validation
- Required fields and default values
- Hierarchical structure

## Python

- Processing data in badges
- Reading metadata from files
- E.g. metadata with data ingestion



You don't need ManGO, these are also standalone applications!

# mango-mdschema

- Offers validation, writing and reading of structured metadata
- Schemas are described in JSON, can be designed in the manager
- Metadata can be hierarchical, rendered with namespacing
- Input can be automatized, output can be parsed and rendered in the portal

# Thank you!

[github.com/kuleuven/mango-mdschema](https://github.com/kuleuven/mango-mdschema)

[github.com/kuleuven/mango-metadata-schemas](https://github.com/kuleuven/mango-metadata-schemas)