

Python package for metadata schemas

Mariana Montes and Ronny Moreas

2024-05-29



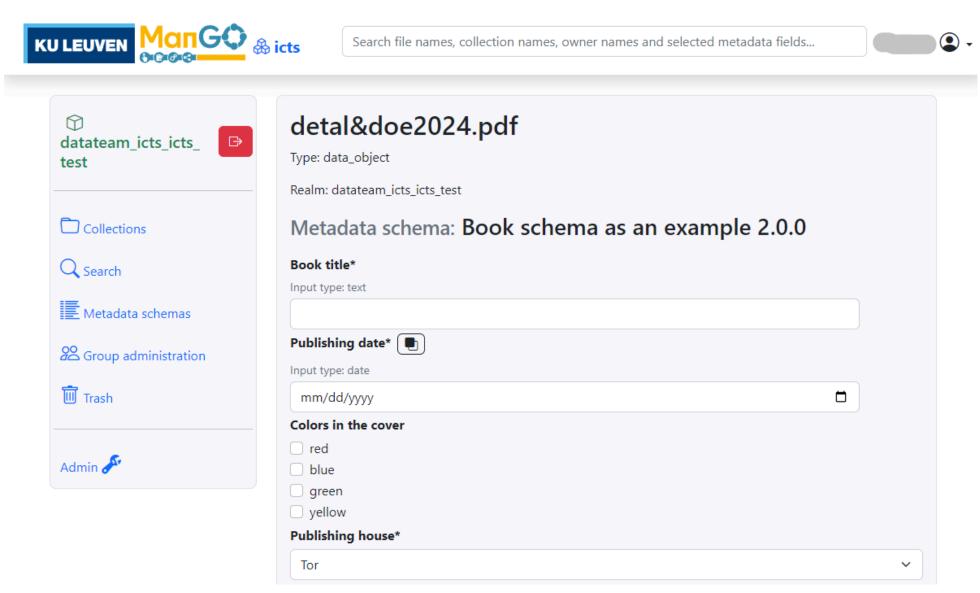
Outline

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

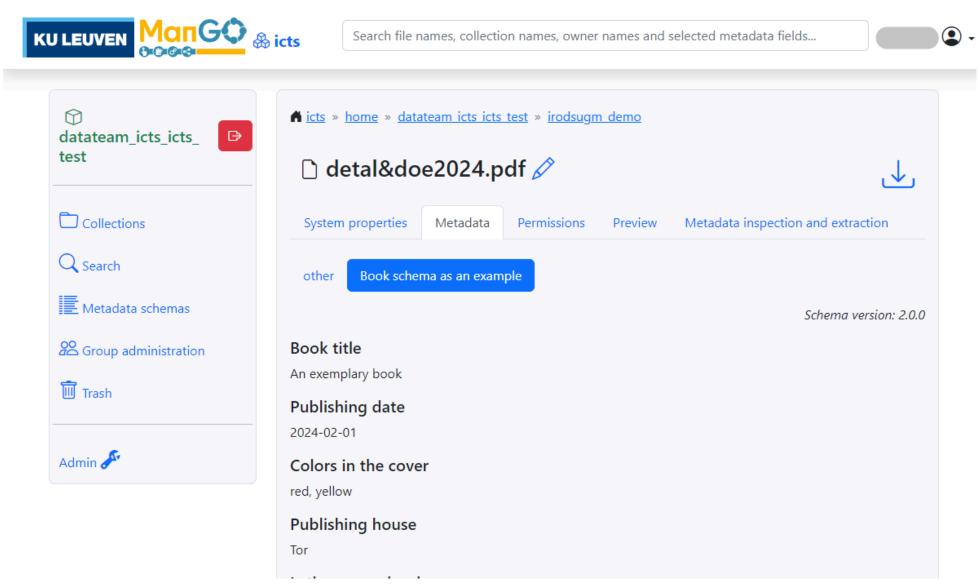
The ManGO Metadata Schema Manager



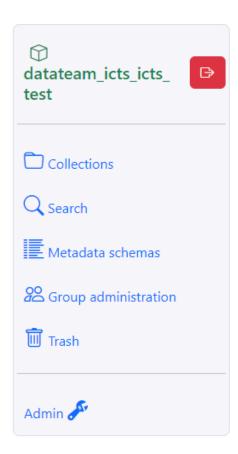
Form to add metadata

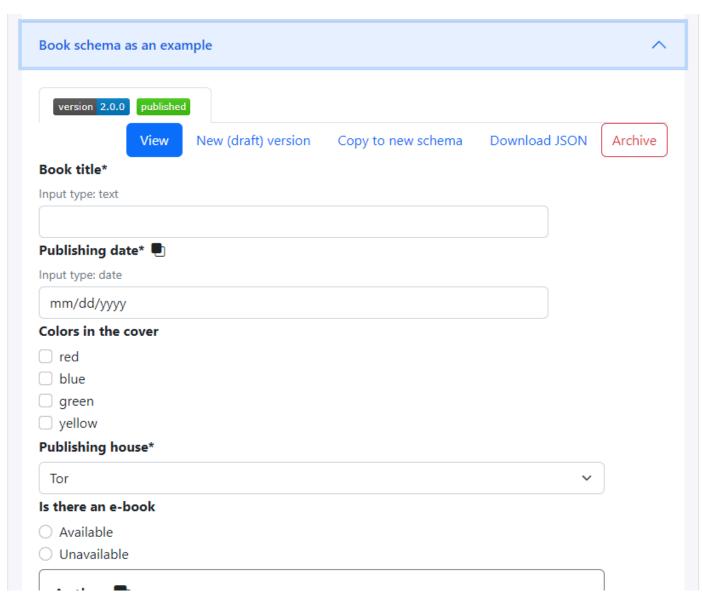


View of the schema metadata



The Schema Manager





Metadata Schemas as JSON

book-v2.0.0-published.json

```
{ 'schema name': 'book',
      'version': '2.0.0',
      'status': 'published',
      'properties': { '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'},
            The sale of the sale of the first and the first through the sale that the sale of the sale
```

Minimal example

1 pip install mango-mdschema

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': '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'},
            The sale of the sale of the first and the first through the sale that the sale of the sale
```

Interpretation via the Python package

```
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'.
```

1 book schema = Schema("book-v2.0.0-published.json")



(1)

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

```
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": "2024-02-01",
                                                                              3
                "cover colors": ["red", "magenta", "yellow", "turquoise"],
        10 }
                                                                               (5)
           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

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

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':
```

```
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 namespacing

```
1 irods object = session.collections.get(home dir).data objects[0]
           irods object.metadata.items()
         1 avus = book schema.to avus(my metadata)
           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 namespacing

Read: from AVUs back to dictionaries

Conclusion



Metadata schemas with Python Metadata schemas Python

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

- 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 github.com/kuleuven/mango-metadata-schemas

