



Structured data

yuuv^{is}® Momentum



Was sind “structured data” in yuuvis Momentum?

- als JSON strukturierte Metadaten
- keine Aufteilung der Daten auf einzelnen Felder notwendig
- kein Schema für die Struktur der JSON-Daten
- dynamische Daten und Strukturen möglich
 - = Vor- und Nachteil!
- starke Suchmöglichkeiten
 - Suche auch in unterschiedlichen Strukturen
 - kombinierte Suche mit anderen Feldtypen
 - Wildcards

Beispiele

- BibJSON (<http://okfnlabs.org/bibjson/>)

```
{
  "title": "Open Bibliography for Science, Technology and
  Medicine",
  "author": [
    {"name": "Richard Jones"},
    {"name": "Mark MacGillivray"},
    {"name": "Peter Murray-Rust"},
    {"name": "Jim Pitman"},
    {"name": "Peter Sefton"},
    {"name": "Ben O'Steen"},
    {"name": "William Waites"}
  ],
  "type": "article",
  "year": "2011",
  "journal": {"name": "Journal of Cheminformatics"},
  "link": [{"url": "http://www.jcheminf.com/content/3/1/47"}],
  "identifier": [{"type": "doi", "id": "10.1186/1758-2946-3-47"}]
}
```

```
# an example collection
{
  "metadata": {
    "collection": "my_collection",
    "label": "My collection of records",
    "description": "a great collection",
    "id": "long_complex_uuid",
    "owner": "test",
    "created": "2011-10-31T16:05:23.055882",
    "modified": "2011-10-31T16:05:23.055882",
    "source": "http://webaddress.com/collection.bib",
    "records": 1594,
    "from": 0,
    "size": 2,
  },
  "records": [
    {
      "collection": "my_collection",
      "type": "book",
      "title": "a great book",
      "id": "your_record_id",
      ...
    },
    ...
  ]
}
```

```
{
  "type": "article",
  "title": "On a family of symmetric Bernoulli
  convolutions",
  "author": [
    {
      "name": "Erdős, Paul"
    }
  ],
  "journal": {
    "name": "American Journal of Mathematics"
    "identifier": [
      {
        "id": "0002-9327",
        "type": "issn"
      }
    ],
    "volume": "61",
    "pages": "974--976"
  },
  "year": "1939",
  "owner": "me",
  "id": "ID_1",
  "collection": "my_collection",
  "url": "http://example.com/me/my_collection/ID_1",
  "link": [
    {
      "url": "http://okfn.org",
      "anchor": "Open Knowledge Foundation"
    }
  ]
}
```



Definition im Schema

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <schema xmlns="http://optimal-systems.org/ns/dmscloud/schema/v5.0/"
3         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4         xsi:schemaLocation="http://optimal-systems.org/ns/dmscloud/schema/v5.0/ dmsCloud-schema.xsd">
5
6     <propertyStructuredDataDefinition>
7         <id>bibjsonsampl:bibjson</id>
8         <description>field for the BibJSON structure identifying and describing the medium</description>
9         <propertyType>structureddata</propertyType>
10        <cardinality>single</cardinality>
11        <required>>false</required>
12    </propertyStructuredDataDefinition>
13    <propertyStringDefinition>
14        <id>bibjsonsampl:locations</id>
15        <description>field for storage locations of printed versions of the medium</description>
16        <propertyType>string</propertyType>
17        <cardinality>multi</cardinality>
18        <required>>false</required>
19    </propertyStringDefinition>
20    <typeDocumentDefinition>
21        <id>bibjsonsampl:medium</id>
22        <baseId>system:document</baseId>
23        <propertyReference>bibjsonsampl:bibjson</propertyReference>
24        <propertyReference>bibjsonsampl:locations</propertyReference>
25        <contentStreamAllowed>allowed</contentStreamAllowed>
26    </typeDocumentDefinition>
27
28 </schema>
```



Daten

```
1 {
2   "type": "book",
3   "title": "My Book Title",
4   "author": [
5     {
6       "name": "Heinrich Schuetzel"
7     },
8     {
9       "name": "Maximilian Sturz"
10    }
11  ],
12  "year": "1995",
13  "owner": "My Library",
14  "id": "ID_of_book",
15  "url": "http://mylibrary.com/ebooks/36513466534",
16  "publisher": "Example Verlag Mustershagen",
17  "edition": {
18    "number": 2,
19    "language": "English"
20  },
21  "identifier": [
22    {
23      "id": "0002-9327",
24      "type": "issn"
25    }
26  ]
27 }
```

```
1 {
2   "metadata": {
3     "collection": "ym_concepts",
4     "description": "Documentation for yuuviz(R) Momentum Core in a struct",
5     "id": "ID_of_collection",
6     "owner": "OS",
7     "created": "2020-06-09T15:04:12.944Z",
8     "modified": "2021-06-10T11:05:17.166Z",
9     "source": "https://help.optimal-systems.com/yuuviz\_develop/display/YM",
10    "records": 2
11  },
12  "records": [
13    {
14      "collection": "ym_concepts",
15      "type": "article",
16      "title": "Schema - Defining Object Types",
17      "id": "ID_of_schema_article",
18      "author": [
19        {
20          "name": "Heinrich Schuetzel"
21        },
22        {
23          "name": "George Trader"
24        },
25        {
26          "name": "Johann Fluss"
27        }
28      ],
29      "created": "2020-06-09T15:04:12.944Z",
30      "modified": "2021-06-10T08:53:23.532Z"
31    },
32    {
33      "collection": "ym_concepts",
34      "type": "article",
35      "title": "Search Query Language",
36    }
37  ]
38 }
```



Verwendungsbeispiele

- Suche nach Dokumenten anhand der Daten in „structured data“-Properties

```
SELECT * FROM appBibjsonsamplesample:medium WHERE appBibjsonsamplesample:bibjson.type = 'book,
```

- Suche nach bestimmten Teilen der „structured data“

```
SELECT appBibjsonsamplesample:bibjson.title FROM appBibjsonsamplesample:medium WHERE appBibjsonsamplesample:bibjson.type = 'book,
```

- Suche nach Werten in verschiedenen Strukturen (**author.name** auf unterschiedlichen Ebenen)

```
SELECT appBibjsonsamplesample:bibjson FROM appBibjsonsamplesample:medium WHERE appBibjsonsamplesample:bibjson..author[*].name CONTAINS('Schuetzel')
```

- Gemischte Suche mit Propertys

```
SELECT * FROM appBibjsonsamplesample:medium WHERE appBibjsonsamplesample:bibjson.type = 'book' AND appBibjsonsamplesample:locations CONTAINS('central')
```



Einschränkungen

- nur eine structured data-Property je Objekt
- Maximal 500 Elemente in den strukturierten Daten eines Objektes
- Unterstützte Typen im JSON
 - String
 - Double
 - Boolean
 - kein Datum, wird als String interpretiert
- Keys müssen den Property-Konventionen entsprechend

Danke

Andreas Dunkel

yuuvis.com



Headquarters · Cicerostraße 26 · 10709 Berlin · Germany
Phone: +49 30 895708-0 · info@yuuvis.com