Mitglied der Helmholtz-Gemeinschaft

JÜLICH FORSCHUNGSZENTRUM

Authority Control in Invenio

[sic!] Jülich – 2nd Invenio User Group Workshop

Jülich, 19. November 2013 | Alexander Wagner, for the Collaboration

Overview

- Why Authority Control?
- Requirements and HowTo
- Authority Records
- Websubmit implementation
- Lacking features

Why Authority Control?

JuSER and friends are the basis for

- Scientific reporting
- Evaluations / Bibliometrics (link up with WoS/Scopus/Inspire)
- Publication lists (e. g. on the WWW: institutes, people, projects)

Why Authority Control?

JuSER and friends are the basis for

- Scientific reporting
- Evaluations / Bibliometrics (link up with WoS/Scopus/Inspire)
- Publication lists (e. g. on the WWW: institutes, people, projects)

All this *needs precise answers* to complex questions:

Normalize as much as possible

Why Authority Control?

JuSER and friends are the basis for

- Scientific reporting
- Evaluations / Bibliometrics (link up with WoS/Scopus/Inspire)
- Publication lists (e. g. on the WWW: institutes, people, projects)

All this *needs precise answers* to complex questions:

Normalize as much as possible: in libraries = Authority Records

Why Authority Control?

JuSER and friends are the basis for

- Scientific reporting
- Evaluations / Bibliometrics (link up with WoS/Scopus/Inspire)
- Publication lists (e. g. on the WWW: institutes, people, projects)

All this *needs precise answers* to complex questions:

Normalize as much as possible: in libraries = Authority Records

Example: History of an institutes name at Jülich

Why Authority Control?

JuSER and friends are the basis for

- Scientific reporting
- Evaluations / Bibliometrics (link up with WoS/Scopus/Inspire)
- Publication lists (e. g. on the WWW: institutes, people, projects)

All this *needs precise answers* to complex questions:

Normalize as much as possible: in libraries = Authority Records

Example: History of an institutes name at Jülich

- till 07/31/2002: ICG-4 (Erdöl und Geochemie)
- till 12/31/2006: ICG-IV (Agrosphäre)
- till 10/31/2010: ICG-4 (Agrosphäre)
- since: IBG-3 (Agrosphäre)

Agrosphäre (ICG-4)

ID I:(DE-Juel1)VDB793

→ ⇦ Agrosphäre → ⇧ ICG ⇧ → Agrosphäre ⇨

Recent Publications

→ All known publications

Book
→ Obersörster, C.
→ **Hydrological characterization of a forest soil using Electrical Resistivity Tomography**
Schriften des Forschungszentrums Jülich. Reihe Energie und Umwelt / Energy und Environment **76,** XXI, 151 S. (2010)

Conference Presentation
→ Rings, J. ; → Scherler, M. ; → Hauck, C.
→ **Coupled thermal and geophysical modelling for monitoring of permafrost**
EGU General Assembly 2010
Vienna, 2010-05-02 (2010)

Conference Presentation

Why not knowledge bases?

- Amount of data (e. g. >30.000 journals, cf. Inspire: 3400)

Why not knowledge bases?

- Amount of data (e. g. >30.000 journals, cf. Inspire: 3400)
- Interchange of records (e. g. Journals take a week to generate $\Rightarrow$ MARC OAI-PMH)

Why not knowledge bases?

- Amount of data (e. g. >30.000 journals, cf. Inspire: 3400)
- Interchange of records (e. g. Journals take a week to generate $\Rightarrow$ MARC OAI-PMH)
- Complex structures (e. g. several linking types)

Why not knowledge bases?

- Amount of data (e. g. >30.000 journals, cf. Inspire: 3400)
- Interchange of records (e. g. Journals take a week to generate ⇒ MARC OAI-PMH)
- Complex structures (e. g. several linking types)
- Flexibility (e. g. additional layers, ignore for selection by `751_7: _IGN_4`, add new fields)

Why not knowledge bases?

- Amount of data (e. g. >30.000 journals, cf. Inspire: 3400)
- Interchange of records (e. g. Journals take a week to generate ⇒ MARC OAI-PMH)
- Complex structures (e. g. several linking types)
- Flexibility (e. g. additional layers, ignore for selection by `751_7: _IGN_4`, add new fields)
- Simplify JavaScript code

Why not knowledge bases?

- Amount of data (e. g. >30.000 journals, cf. Inspire: 3400)
- Interchange of records (e. g. Journals take a week to generate ⇒ MARC OAI-PMH)
- Complex structures (e. g. several linking types)
- Flexibility (e. g. additional layers, ignore for selection by `751_7: _IGN_4`, add new fields)
- Simplify JavaScript code
- Reuse existing records (e. g. German National Library, LoC. . .)
- Automatic ingestion (e. g. convert LDAP to MARC)

Why not knowledge bases?

- Amount of data (e. g. >30.000 journals, cf. Inspire: 3400)
- Interchange of records (e. g. Journals take a week to generate ⇒ MARC OAI-PMH)
- Complex structures (e. g. several linking types)
- Flexibility (e. g. additional layers, ignore for selection by `751_7: _IGN_4`, add new fields)
- Simplify JavaScript code
- Reuse existing records (e. g. German National Library, LoC. . .)
- Automatic ingestion (e. g. convert LDAP to MARC)

But mainly...

Why not knowledge bases?

- Amount of data (e. g. >30.000 journals, cf. Inspire: 3400)
- Interchange of records (e. g. Journals take a week to generate ⇒ MARC OAI-PMH)
- Complex structures (e. g. several linking types)
- Flexibility (e. g. additional layers, ignore for selection by `751_7: _IGN_4`, add new fields)
- Simplify JavaScript code
- Reuse existing records (e. g. German National Library, LoC. . .)
- Automatic ingestion (e. g. convert LDAP to MARC)

But mainly...

Librarians are comfortable with MARC, but not kb

Why not knowledge bases?

- Amount of data (e. g. >30.000 journals, cf. Inspire: 3400)
- Interchange of records (e. g. Journals take a week to generate ⇒ MARC OAI-PMH)
- Complex structures (e. g. several linking types)
- Flexibility (e. g. additional layers, ignore for selection by `751_7: _IGN_4`, add new fields)
- Simplify JavaScript code
- Reuse existing records (e. g. German National Library, LoC. . .)
- Automatic ingestion (e. g. convert LDAP to MARC)

But mainly...

Librarians are comfortable with MARC, but not kb

Thus, routine work (e. g. typos, linkage. . .) can be handled by librarians and does not require a technician/programmer.

"Dirty" sets from migration need human intervention. . .

Our users – oh, there was something. . .

Example: What happens if you select a journal?

Say, starting out with a DOI, e. g. 10.1103/PhysRevB.87.081302

Our users – oh, there was something...

Example: What happens if you select a journal?

Say, starting out with a DOI, e. g. 10.1103/PhysRevB.87.081302

From our authorities you get for free:

```
0247_ $2ISSN$a0163-1829
0247_ $2ISSN$a1095-3795
0247_ $2ISSN$a1550-235X
0247_ $2ISSN$a0556-2805
0247_ $2ISSN$a1098-0121
082__ $a530
260__ $aCollege Park, Md.$bAPS
773__ $0PERI:(DE-600)1473011-x$a10.1103/PhysRevB.87.081302...
915__ $0StatID:(DE-HGF)0100$2StatID$aJCR
915__ $0StatID:(DE-HGF)0200$2StatID$aDBCoverage$bSCOPUS
915__ $0StatID:(DE-HGF)0300$2StatID$aDBCoverage$bMedline
915__ $0StatID:(DE-HGF)0420$2StatID$aNationallizenz
915__ $0StatID:(DE-HGF)0110$2StatID$aWoS$bScience Citation Index
915__ ...
```

Our users – oh, there was something. . .

Example: What happens if you select a journal?

Say, starting out with a DOI, e. g. 10.1103/PhysRevB.87.081302

From our authorities you get for free:

```
0247_ $2ISSN$a0163-1829
0247_ $2ISSN$a1095-3795
0247_ $2ISSN$a1550-235X
0247_ $2ISSN$a0556-2805
0247_ $2ISSN$a1098-0121
082__ $a530
260__ $aCollege Park, Md.$bAPS
773__ $0PERI:(DE-600)1473011-x$a10.1103/PhysRevB.87.081302...
915__ $0StatID:(DE-HGF)0100$2StatID$aJCR
915__ $0StatID:(DE-HGF)0200$2StatID$aDBCoverage$bSCOPUS
915__ $0StatID:(DE-HGF)0300$2StatID$aDBCoverage$bMedline
915__ $0StatID:(DE-HGF)0420$2StatID$aNationallizenz
915__ $0StatID:(DE-HGF)0110$2StatID$aWoS$bScience Citation Index
915__ ...
```

Did you/your secretary know. . .

. . . e. g. that Phys. Rev. B is in Medline and has DDC 530?

For our queries

Authorities allows us to:

- find all publications in a given journal
- find all publications in journals listed in a given database
- find all publications in a given field
- do the reporting:
  - "What articles are in WoS or Scopus?" — "And Pubmed?"
  - "Which of our OpenAccess articles have an Impact Factor?"
  - "What did we publish in Nature, no not the Baby-Natures?"
- do quite unique queries (e. g. all OpenAccess Journals that have an Impact factor)

For our queries

Authorities allows us to:

- find all publications in a given journal
- find all publications in journals listed in a given database
- find all publications in a given field
- do the reporting:
  - “What articles are in WoS or Scopus?” — “And Pubmed?”
  - “Which of our OpenAccess articles have an Impact Factor?”
  - “What did we publish in Nature, no not the Baby-Natures?”
- do quite unique queries (e. g. all OpenAccess Journals that have an Impact factor)

But basically. . .

we get better, more consistent data without our users need to enter them manually each time.

Authority Records: Requirements

1. n:m relations (e. g. splitting of institutes, grants with several top levels)
2. Tracking history (predecessors and successors)
3. Tracking hierarchy (dad/son relations; we could have several parents!)
4. Several identifiers (e. g. DOI + Handle + URN or ORCiD + other IDs)
5. Multiple types of authorities (e. g. people, grants, institutes, journals. . .)

Authority Records: Requirements

1. n:m relations (e. g. splitting of institutes, grants with several top levels)
2. Tracking history (predecessors and successors)
3. Tracking hierarchy (dad/son relations; we could have several parents!)
4. Several identifiers (e. g. DOI + Handle + URN or ORCiD + other IDs)
5. Multiple types of authorities (e. g. people, grants, institutes, journals. . .)

Solved by MARC Authority

Authority Records: Requirements

1. n:m relations (e. g. splitting of institutes, grants with several top levels)
2. Tracking history (predecessors and successors)
3. Tracking hierarchy (dad/son relations; we could have several parents!)
4. Several identifiers (e. g. DOI + Handle + URN or ORCiD + other IDs)
5. Multiple types of authorities (e. g. people, grants, institutes, journals. . .)

Solved by MARC Authority

1. repeatable linking fields (MARC Authority 4xx, 5xx)

Authority Records: Requirements

1. n:m relations (e. g. splitting of institutes, grants with several top levels)
2. Tracking history (predecessors and successors)
3. Tracking hierarchy (dad/son relations; we could have several parents!)
4. Several identifiers (e. g. DOI + Handle + URN or ORCiD + other IDs)
5. Multiple types of authorities (e. g. people, grants, institutes, journals. . .)

Solved by MARC Authority

1. repeatable linking fields (MARC Authority 4xx, 5xx)
2. horizontal linking ($w control subfield: $wa, $wb)

Authority Records: Requirements

1. n:m relations (e. g. splitting of institutes, grants with several top levels)
2. Tracking history (predecessors and successors)
3. Tracking hierarchy (dad/son relations; we could have several parents!)
4. Several identifiers (e. g. DOI + Handle + URN or ORCiD + other IDs)
5. Multiple types of authorities (e. g. people, grants, institutes, journals. . .)

Solved by MARC Authority

1. repeatable linking fields (MARC Authority 4xx, 5xx)
2. horizontal linking ($w control subfield: $wa, $wb)
3. vertical linking ($w control subfield: $wt)

Authority Records: Requirements

1. n:m relations (e. g. splitting of institutes, grants with several top levels)
2. Tracking history (predecessors and successors)
3. Tracking hierarchy (dad/son relations; we could have several parents!)
4. Several identifiers (e. g. DOI + Handle + URN or ORCiD + other IDs)
5. Multiple types of authorities (e. g. people, grants, institutes, journals. . .)

Solved by MARC Authority

1. repeatable linking fields (MARC Authority 4xx, 5xx)
2. horizontal linking ($w control subfield: $wa, $wb)
3. vertical linking ($w control subfield: $wt)
4. `0247_` with $2 = source and/vs. `035__` (we do not repeat 035)

Authority Records: Requirements

1. n:m relations (e. g. splitting of institutes, grants with several top levels)
2. Tracking history (predecessors and successors)
3. Tracking hierarchy (dad/son relations; we could have several parents!)
4. Several identifiers (e. g. DOI + Handle + URN or ORCiD + other IDs)
5. Multiple types of authorities (e. g. people, grants, institutes, journals. . .)

Solved by MARC Authority

1. repeatable linking fields (MARC Authority 4xx, 5xx)
2. horizontal linking ($w control subfield: $wa, $wb)
3. vertical linking ($w control subfield: $wt)
4. `0247_` with $2 = source and/vs. `035__` (we do not repeat 035)
5. ignorant of the type but specify source (usually: $0 or $a and $2 subfields)

Where to use?

- Author disambiguation (ready for ORCiD)
- Institutes (history and hierarchy)
- Grants (history and hierarchy, also: several sources)
- Journals (though bibliographic by their nature)
- Statistic Keys (e. g. "has impact factor", "is listed in pubmed", "is OpenAccess")
- Vocabulary (e. g. level within a hierarchy)
- Licence information (e. g. CC-BY)
- . . . (Due to its flexibility we can adopt easily for new use cases)

Where to use?

- Author disambiguation (ready for ORCiD)
- Institutes (history and hierarchy)
- Grants (history and hierarchy, also: several sources)
- Journals (though bibliographic by their nature)
- Statistic Keys (e. g. “has impact factor”, “is listed in pubmed”, “is OpenAccess”)
- Vocabulary (e. g. level within a hierarchy)
- Licence information (e. g. CC-BY)
- . . . (Due to its flexibility we can adopt easily for new use cases)

Define:

- MARC datasets (automatize creation, or manually)
- output formats (HD, HB, JS = JSON)

Authority Records HowTo

Looks (almost) like MARC bibliographic:

```
001__ 98676
1101_ $aAgrosphäre$y-31.10.2010
4101_ $wd$aICG-4
500__ $aRecord converted from VDB: 14.11.2012
909CO $ooai:juser.fz-juelich.de:98676$pauthority
970__ $aI:(DE-Juel1)VDB793
980__ $aI
980__ $aAUTHORITY
```

Authority Records HowTo

Looks (almost) like MARC bibliographic:

```
001__ 98676
1101_ $aAgrosphäre$y-31.10.2010
4101_ $wd$aICG-4
500__ $aRecord converted from VDB: 14.11.2012
909CO $ooai:juser.fz-juelich.de:98676$pauthority
970__ $aI:(DE-Juel1)VDB793
980__ $aI
980__ $aAUTHORITY
```

- Add identifier(s): (Note: *(DE-Juel1)* allows for interchage)

```
0247_ $aI:(DE-Juel1)VDB793$2I:(DE-Juel1)
0247_ $a2128/25$2Handle
035__ $aI:(DE-Juel1)VDB793
```

Authority Records HowTo

Looks (almost) like MARC bibliographic:

```
001__ 98676
1101_ $aAgrosphäre$y-31.10.2010
4101_ $wd$aICG-4
500__ $aRecord converted from VDB: 14.11.2012
909CO $ooai:juser.fz-juelich.de:98676$pauthority
970__ $aI:(DE-Juel1)VDB793
980__ $aI
980__ $aAUTHORITY
```

- Add identifier(s): (Note: *(DE-Juel1)* allows for interchage)

```
0247_ $aI:(DE-Juel1)VDB793$2I:(DE-Juel1)
0247_ $a2128/25$2Handle
035__ $aI:(DE-Juel1)VDB793
```

- Add horizontal and vertical links:

```
5101_ $wa$0I:(DE-Juel1)VDB50$2I:(DE-Juel1)$aAgrosphäre
5101_ $wb$0I:(DE-Juel1)IBG-3-20101118$2I:(DE-Juel1)$aAgrosphäre
5101_ $wt$0I:(DE-Juel1)VDB1109$2I:(DE-Juel1)$aICG
```

(see also Library of Congress)

And Websubmit?

JSON output formats for Ajax/JavaScript

Note: we use MARC tags – ambiguous “human readable names” are not helpful. . .

And Websubmit?

JSON output formats for Ajax/JavaScript

Note: we use MARC tags – ambiguous “human readable names” are not helpful. . .

- Institutes


```
{ ''I9201_0'' : ''I:(DE-Juel1)VDB793'',
''I9201_l'' : ''Agrosphäre'',
''I9201_k'' : ''ICG-4'',
''I980__a'' : ''I:(DE-Juel1)VDB793'',
''label'' : ''ICG-4: Agrosphäre'' }
```

And Websubmit?

JSON output formats for Ajax/JavaScript

Note: we use MARC tags – ambiguous “human readable names” are not helpful. . .

- Institutes

```
{ ”I9201_0” : ”I:(DE-Juel1)VDB793”,
”I9201_l” : ”Agrosphäre”,
”I9201_k” : ”ICG-4”,
”I980__a” : ”I:(DE-Juel1)VDB793”,
”label” : ”ICG-4: Agrosphäre” }
```

- Journals

```
{ ”I773__t” : ”Physical review / D”,
”I773__0” : ”PERI:(DE-600)1461167-3”,
”I773__x” : ”1089-4918”,
”I082__a” : ”530”,
”I260__a” : ”New York, NY”,
”I260__b” : ”Soc.”,
”I915__” : ”[{<stat keys dict 1>}, {<stat keys dict 2>}...]”

”label” : ”Physical review / D (1089-4918)” }
```

Sidenote: OpenAire is just the same...

OpenAire = EU Funding:

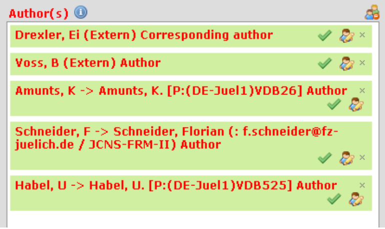
{ ''I536__0'': ''G:(EU-Grant)269921'',
''I536__a'': ''BRAINSCALES - Brain-inspired multiscale computation in neuromorphic hybrid systems (269921)'',
''I536__c'': ''269921'',
''I536__f'': ''FP7-ICT-2009-6'',
''I9131_0'': ''G:(EU-Grant)269921'',
''I9131_a'': ''DE-HGF'',
''I9131_v'': ''Brain-inspired multiscale computation in neuromorphic hybrid systems'',
''label'': ''BRAINSCALES - Brain-inspired multiscale computation in neuromorphic hybrid systems (2010-12-31 - 2014-12-31)'' },

Local funding scheme (POF):

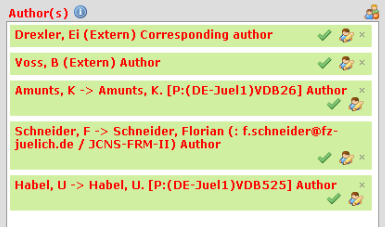
{ ''I536__0'': ''G:(DE-HGF)POF2-111'',
''I536__a'': ''111 - Thin Film Photovoltaics (POF2-111)'',
''I536__c'': ''POF2-111'',
''I536__f'': ''POF II'',
''I9131_0'': ''G:(DE-HGF)POF2-111'',
''I9131_1'': ''G:(DE-HGF)POF2-110'',
''I9131_2'': ''G:(DE-HGF)POF2-100'',
''I9131_a'': ''DE-HGF'',
''I9131_b'': ''Energie'',
''I9131_l'': ''Erneuerbare Energien'',
''I9131_v'': ''Thin Film Photovoltaics'',

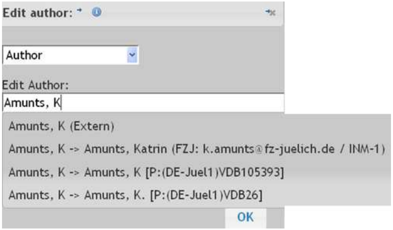
''label'': ''111 - Thin Film Photovoltaics (POF II: 2010 - 2014)'' },

How does it look like?

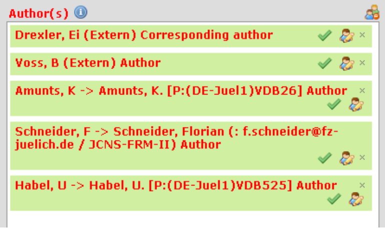


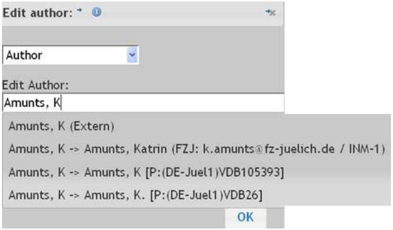
How does it look like?





How does it look like?

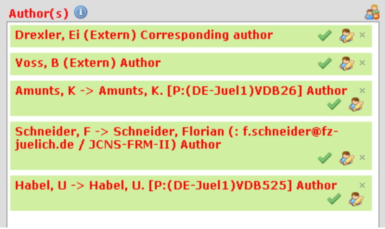


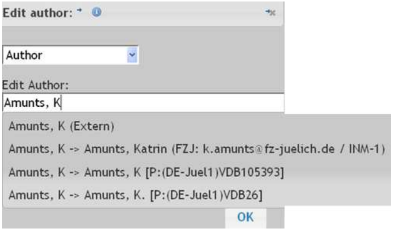


Backend:

- http-queries requesting `&of=JS` (Invenio is fast enough!)

How does it look like?

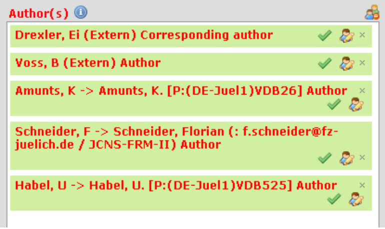


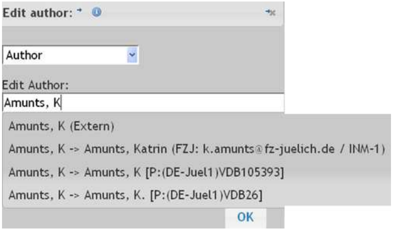


Backend:

- http-queries requesting `&of=JS` (Invenio is fast enough!)
- `bibreformat -o JS` (just speedup, at some points we need recursions. . .)

How does it look like?

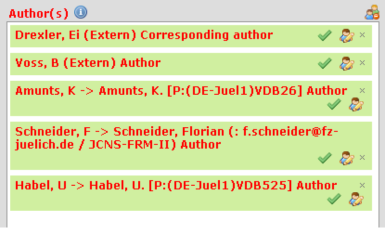


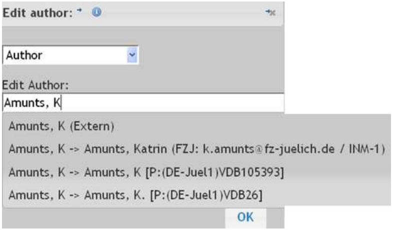


Backend:

- http-queries requesting `&of=JS` (Invenio is fast enough!)
- `bibreformat -o JS` (just speedup, at some points we need recursions. . .)
- washing routines (output formats don't allow for list prefix/suffixes)

How does it look like?

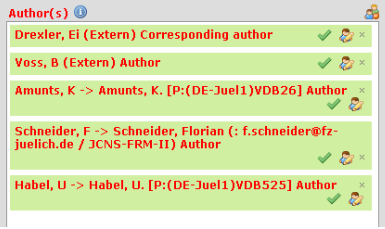


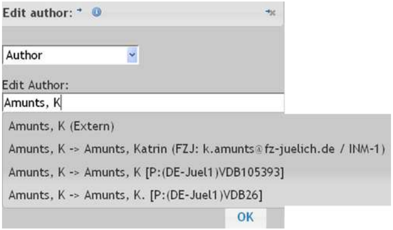


Backend:

- http-queries requesting `&of=JS` (Invenio is fast enough!)
- `bibreformat -o JS` (just speedup, at some points we need recursions. . .)
- washing routines (output formats don't allow for list prefix/suffixes)
- few pythonic output formats (most could be clicked in web GUI, however: don't do that)

How does it look like?





Backend:

- http-queries requesting `&of=JS` (Invenio is fast enough!)
- `bibreformat -o JS` (just speedup, at some points we need recursions. . .)
- washing routines (output formats don't allow for list prefix/suffixes)
- few pythonic output formats (most could be clicked in web GUI, however: don't do that)
- sometimes pythonic helpers (if it gets complex: author guessing, recursions. . .)

Lacking features

- Autority-based searches
  - with children/tops
  - with predecessors/successors
  - `noexp` parameter to search without explosion

Lacking features

- Autority-based searches
  - with children/tops
  - with predecessors/successors
  - `noexp` parameter to search without explosion
- Handle multiple IDs

Lacking features

- Autority-based searches
  - with children/tops
  - with predecessors/successors
  - `noexp` parameter to search without explosion
- Handle multiple IDs

```
001__ 95749
0247_ $aP:(DE-Juel1)133832$2P:(DE-Juel1)
0247_ $a0000-0001-9846-5516$2ORCID
0247_ $aI-3159-2013$2ResearcherID
035__ $aP:(DE-Juel1)133832
1001_ $aWagner, Alexander$gmale
...
750_7 $0V:(DE-HGF)8$2HGFVOC$aAuthorized$d2013-11-09
980__ $aP
980__ $aAUTHORITY
```

Lacking features

- Autority-based searches
  - with children/tops
  - with predecessors/successors
  - `noexp` parameter to search without explosion
- Handle multiple IDs

```
001__ 95749
0247_ $aP:(DE-Juel1)133832$2P:(DE-Juel1)
0247_ $a0000-0001-9846-5516$2ORCID
0247_ $aI-3159-2013$2ResearcherID
035__ $aP:(DE-Juel1)133832
1001_ $aWagner, Alexander$gmale
...
750_7 $0V:(DE-HGF)8$2HGFVOC$aAuthorized$d2013-11-09
980__ $aP
980__ $aAUTHORITY
```

- Lookup IDs for searches (e. g. insert “IBG-3”, get ID search for cid:”<ID>”)

Lacking features

- Autority-based searches
  - with children/tops
  - with predecessors/successors
  - `noexp` parameter to search without explosion
- Handle multiple IDs

```
001__ 95749
0247_ $aP:(DE-Juel1)133832$2P:(DE-Juel1)
0247_ $a0000-0001-9846-5516$2ORCID
0247_ $aI-3159-2013$2ResearcherID
035__ $aP:(DE-Juel1)133832
1001_ $aWagner, Alexander$gmale
...
750_7 $0V:(DE-HGF)8$2HGFVOC$aAuthorized$d2013-11-09
980__ $aP
980__ $aAUTHORITY
```

- Lookup IDs for searches (e. g. insert “IBG-3”, get ID search for cid:”<ID>”)

Basically, search functionality is lacking to get the most out of it

Current work arrounds

- Predecessors:
  Add all previous IDs to a certain MARC-field (e. g. `920` + `981`) and search in an index (e. g. `cid`) that collects these fields.
- Statistics:
  Use keys for search and in case loop. Works, but might be time consuming.
- Introduction of ORCiD:
  Probably need to write several Ids for one author, as personal publication lists are based on current Ids.

Hopefully, all this is solved by upcoming authority search.

Thanks!

Alexander Wagner
Zentralbibliothek

Scientific Services /
Scientific Publishing

Tel.: +49-2461-61-1586
a.wagner@fz-juelich.de

This document is available as FZJ-2013-05411

Typeset by pdfLaTeX