Mitglied der Helmholtz-Gemeinschaft

JÜLICH FORSCHUNGSZENTRUM

Invenio @ HGF

JuSER

June 9, 2013 | Alexander Wagner

Overview

- History
- JuSER: Display, document flow
- Submission of Documents
- Author handling
- Lorg
- Personalization

JÜLICH FORSCHUNGSZENTRUM

Invenio @ HGF

Part I: History

June 9, 2013 | Alexander Wagner

VDB and JUWEL

Publications database VDB:

- Centralized database of the scientific output
- $>$ 62.000 entries from $\approx$ 13 years ($\approx +4800/a$)
- Obligatory input[1] (editors at each institute, crosschecked by ZB)

[1] cf. Publication Guidelines

VDB and JUWEL

Publications database VDB:

- Centralized database of the scientific output
- > 62.000 entries from $\approx$ 13 years ($\approx +4800/a$)
- Obligatory input[1] (editors at each institute, crosschecked by ZB)
- Basis for
  - Scientific Report
  - Evaluations
  - Publication lists (e. g. WWW)

[1] cf. Publication Guidelines

VDB and JUWEL

Publications database VDB:

- Centralized database of the scientific output
- > 62.000 entries from $\approx$ 13 years ($\approx +4800/a$)
- Obligatory input[1] (editors at each institute, crosschecked by ZB)
- Basis for
  - Scientific Report
  - Evaluations
  - Publication lists (e. g. WWW)

JUWEL:

- Berlin Declaration for Open Access
- Institutional repository ($\approx$ 4500 full text files, $\approx +560/a$)

---

[1] cf. Publication Guidelines

Forschungszentrum Jülich – Publikationen – Iceweasel

File Edit View History Bookmarks Tools Help Search News Bank Uni Bib Comp Cam

w.fz-juelich.de/inm/inm-1/DE/Service/Publikationslisten/publikationslisten_node.html JuSER

Suchbegriff MITARBEITERSUCHE | DEUTSCH | ENGLISH Institutswegweiser

JÜLICH FORSCHUNGSZENTRUM

Institut für Neurowissenschaften und Medizin
Strukturelle und funktionelle Organisation des Gehirns (INM-1)

AKTUELLES FORSCHUNG LEISTUNGEN KARRIERE ÜBER UNS

INM-1 Forschung Publikationen

FORSCHUNG

- Architektonik und Hirnfunktion
- Multimodale Bildverarbeitung
- Modellierung kortikaler Systeme
- Genomic Imaging
- **Publikationen**

SERVICE

- Ansprechpartner
- Mitarbeiter
- Publikationen
- Anfahrt
- Downloads

Zeitschriftenbeiträge 2012

Amunts, K. ; Zilles, K.
Architecture and organizational principles of Brocas region
Trends in cognitive sciences **16,** 418 - 426 (2012) [10.1016/j.tics.2012.06.005]
S·F·X

Bis, J. C. ; et, a.
Common variants at 12q14 and 12q24 are associated with hippocampal volume
Nature genetics **44,** 545 - 551 (2012) [10.1038/ng.2237] S·F·X

Boutros, N. N. ; Gjini, K. ; Eickhoff, S. B. ; Urbach, H. ; Pflieger, M. E.
Mapping repetition suppression of the P50 evoked response to the human cerebral cortex
Clinical neurophysiology **00,** 00 (2012) [10.1016/j.cinph.2012.10.007] S·F·X

In Print

Bzdok, D. ; Laird, A. ; Zilles, K. ; Fox, P. ; Eickhoff, S. B.
An investigation of the structural, connectional, and functional sub-specialization in the human amygdala
Human brain mapping (2012) [10.1002/hbm.22138] S·F·X

Bzdok, D. ; Schilbach, L. ; Vogeley, K. ; Schneider, K. ; Laird, A. R. ; Langner, R. ; Eickhoff, S. B.
Parsing the neural correlates of moral cognition: ALE meta-analysis on morality, theory of mind, and empathy
Brain structure & function **217,** 783 - 796 (2012) [10.1007/s00429-012-0380-y]
S·F·X

PUBLIKATIONEN INM-1

- Publikationen 2011
- Publikationen 2010
- Publikationen 2009
- Publikationen 2008

→ Mehr

Longitudinal deformation-based morphometry reveals spatio-temporal dynamics of brain volume changes in patients with corticobasal syndrome - JuSER - Mozilla Firefox

File Edit View History Bookmarks Tools Help | Most Visited | Search | News | Uni | FR | Bib | Wiki | SRV | Chess | Comp

juser.fz-juelich.de/record/130981 | Startpage (SSL)

admin :: Abmelden

JuSER

JÜLICH FORSCHUNGSZENTRUM

SUCHEN | ABSENDEN | PERSONALISIEREN | HILFE | VERWALTUNG

Hauptseite > Longitudinal deformation-based morphometry reveals spatio-temporal dynamics of brain volume changes in patients with corticobasal syndrome

Information | Diskussion | Dateien | Plots

Journal Article | FZJ-2012-00157

Longitudinal deformation-based morphometry reveals spatio-temporal dynamics of brain volume changes in patients with corticobasal syndrome

Pieperhoff, Peter* ; Ferrea, Stefano ; Krause, Holger ; Groiss, Stefan Jun ; Elben, Saskia ; Wojtecki, Lars ; Zilles, Karl* ; Amunts, Katrin* ; Schnitzler, Alfons ; Südmeier, M (Corresponding author)

2012

PLoS Lawrence, Kan.

Published in: PLoS one 7 (7) e41873 - [10.1371/journal.pone.0041873] S·F·X

Pubmed; OPEN ACCESS Pubmed Central Fulltext;
Web of Science (WOS); Citing articles (WOS); Related articles (WOS)

Report No.: FZJ-2012-00157

Abstract: Corticobasal syndrome (CBS) is a rare neurodegenerative disorder characterized by a progressive and asymmetric manifestation of cortical and basal-ganglia symptoms of different origin.
The spatio-temporal dynamics of cerebral atrophy in CBS is barely known. This study aimed to longitudinally quantify the individual dynamics of brain volume changes in patients with CBS as compared

Contributing Institute(s):

1. Strukturelle und funktionelle Organisation des Gehirns (INM-1)
2. Zentralbibliothek (ZB)

Research Program(s):

1. 332 - Imaging the Living Brain (POF2-332) (Gesundheit)

Appears in the scientific report 2012

Database coverage:

JuSER

- Import interfaces (improve data quality, ease up input)

JuSER

- Import interfaces (improve data quality, ease up input)
- Exports to BibTeX/EndNote (Integrate with citations management)

JuSER

- Import interfaces (improve data quality, ease up input)
- Exports to BibTeX/EndNote (Integrate with citations management)
- Institute collections (collect and share documents)

JuSER

- Import interfaces (improve data quality, ease up input)
- Exports to BibTeX/EndNote (Integrate with citations management)
- Institute collections (collect and share documents)
- Add full texts

JuSER

- Import interfaces (improve data quality, ease up input)
- Exports to BibTeX/EndNote (Integrate with citations management)
- Institute collections (collect and share documents)
- Add full texts
- Normalize as much as possible

JuSER

- Import interfaces (improve data quality, ease up input)
- Exports to BibTeX/EndNote (Integrate with citations management)
- Institute collections (collect and share documents)
- Add full texts
- Normalize as much as possible
  - Authors (tell apart Meier and Meier)
  - Institutes
  - Journals
  - Projects (POF, EU, . . .)
  - . . .

JuSER

- Import interfaces (improve data quality, ease up input)
- Exports to BibTeX/EndNote (Integrate with citations management)
- Institute collections (collect and share documents)
- Add full texts
- Normalize as much as possible
  - Authors (tell apart Meier and Meier)
  - Institutes
  - Journals
  - Projects (POF, EU, . . .)
  - . . .

. . . Users don't need to care about technical details . . .


JuSER - JuSER - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Search News Uni FR Bib Wiki SRV Chess Comp Tiny
juser.fz-juelich.de/?ln=en
JuSER
admin :: logout
JuSER
JÜLICH
FORSCHUNGSZENTRUM
SEARCH SUBMIT PERSONALIZE HELP ADMINISTRATION
Search 128,357 records for:
any field
Search Browse
Search Tips :: Advanced Search
Narrow by collection:
Publications database (64,428)
Documents in print (102)
JUWEL (2,583)
Institute Collections (56,073)
B (0) BFC (0) BR (0) BSG (0) DSB (0) ETN (0) F (0) FA.PARTEC (0) FD (0)
FGZ (0) FS (0) G (0) GRS (0) IAS (1,293) IBG (5,945) IBN (2,184) IBOC (0)
ICS (6,201) IEK (15,949) IKM (0) IKP (3,759) IMET (19) INB (2,982) INM (3,894)
ITS (0) JARA (0) JCNS (1,992) JSC (4,013) JULAB (0) KME (0) M (0) MOD (0)
N (0) NIC (222) O (0) P (0) PGI (11,089) PTJ (0) R (0) REV (0) S (0) SL (0)
T (0) TB (0) UE (0) UK (0) US (0) VB (1) VS (0) WTR (0) ZAT (461) ZB (394)
ZC (0) ZCH (869) ZEL (936)
Authorities (63,269)
Grants (15,266) Institutions (2) Institutes (437) People (18,108) Periodicals (29,391)
Publication types (35) Statistics keys (25) Controlled vocabulary (7)
Focus on:
Document types (64,538)
Articles (17,733) Books (7,429)
Events (75) Other Resources (0)
Patents (2) Presentations (43,026)
Reports (868) Theses (2,138)
Unpublished (0)
QUICK LINKS
Central Library
Library Catalogue
Literature Request
Reading Room
Publishing House
Subject Information Portal
Recent additions
In Print

Collections

- Publicationsdatabase: Publications from Jülich

Collections

- Publicationsdatabase: Publications from Jülich
- Documents in Print:
  - approved by institutes editor
  - not approved by the library
  - e. g. papers just available Online (missing bibliographic data e. g. pages, volume)

Collections

- Publicationsdatabase: Publications from Jülich
- Documents in Print:
  - approved by institutes editor
  - not approved by the library
  - e. g. papers just available Online (missing bibliographic data e. g. pages, volume)
  - already visible on institutes web pages

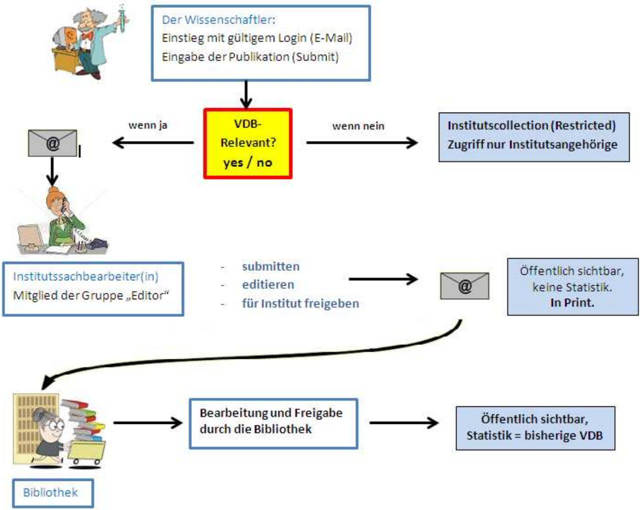
Collections

- Publicationsdatabase: Publications from Jülich
- Documents in Print:
  - approved by institutes editor
  - not approved by the library
  - e. g. papers just available Online (missing bibliographic data e. g. pages, volume)
  - already visible on institutes web pages
- OpenAccess repository JUWEL

Collections

- Publicationsdatabase: Publications from Jülich
- Documents in Print:
  - approved by institutes editor
  - not approved by the library
  - e. g. papers just available Online (missing bibliographic data e. g. pages, volume)
  - already visible on institutes web pages
- OpenAccess repository JUWEL
- Institute Collections:
  - Institutes private workspace
  - e. g. drafts, collected literature, journal clubs ...
  - Access for members of the institue only

Document Flow



Thanks to H. Lexis

Institute Collections

JuSER can hold documents beyond own publications

Institute Collections

JuSER can hold documents beyond own publications

- webbased literature management

Institute Collections

JuSER can hold documents beyond own publications

- webbased literature management
- document exchange at the institutes

Institute Collections

JuSER can hold documents beyond own publications

- webbased literature management
- document exchange at the institutes
- centralized collection of papers

Institute Collections

JuSER can hold documents beyond own publications

- webbased literature management
- document exchange at the institutes
- centralized collection of papers
- easy export to formatting tools (BibTEX, EndNote)

Institute Collections

JuSER can hold documents beyond own publications

- webbased literature management
- document exchange at the institutes
- centralized collection of papers
- easy export to formatting tools (BibTEX, EndNote)
- commenting (individual and in group)

Institute Collections

JuSER can hold documents beyond own publications

- webbased literature management
- document exchange at the institutes
- centralized collection of papers
- easy export to formatting tools (BibTEX, EndNote)
- commenting (individual and in group)

- Institute collections require proper login
- Non-VDB relevant items do not show up on the webpage
- ZB does not care about usage

Mitglied der Helmholtz-Gemeinschaft

JÜLICH FORSCHUNGSZENTRUM

Invenio @ HGF

Part II: Basic Usage

June 9, 2013 | Alexander Wagner

Submit

1. Log in (LDAP based)
2. Select Submit from the main menu (http://juser.fz-juelich.de/submit)
3. Select document type

Submit

1. Log in (LDAP based)
2. Select Submit from the main menu (http://juser.fz-juelich.de/submit)
3. Select document type
4. Fill in the submit form

Submit

1. Log in (LDAP based)
2. Select Submit from the main menu (http://juser.fz-juelich.de/submit)
3. Select document type
4. Fill in the submit form
   - Use import if possible (doi, pmid, arXiv...)
   - take care of the authors
   - Own publication? $\Rightarrow$ VDB-Relevant = yes

Submit

1. Log in (LDAP based)
2. Select Submit from the main menu (http://juser.fz-juelich.de/submit)
3. Select document type
4. Fill in the submit form
   - Use import if possible (doi, pmid, arXiv. . .)
   - take care of the authors
   - Own publication? ⇒ VDB-Relevant = yes
5. Add full text
6. Submit it:

Submit

1. Log in (LDAP based)
2. Select Submit from the main menu (http://juser.fz-juelich.de/submit)
3. Select document type
4. Fill in the submit form
   - Use import if possible (doi, pmid, arXiv. . .)
   - take care of the authors
   - Own publication? ⇒ VDB-Relevant = yes
5. Add full text
6. Submit it:
   - “Finish & Release”: proceed to next level
   - “Postpone”: generates `TEMPENTRY`

Submit

1. Log in (LDAP based)
2. Select Submit from the main menu (http://juser.fz-juelich.de/submit)
3. Select document type
4. Fill in the submit form
   - Use import if possible (doi, pmid, arXiv. . .)
   - take care of the authors
   - Own publication? ⇒ VDB-Relevant = yes
5. Add full text
6. Submit it:
   - "Finish & Release": proceed to next level
   - "Postpone": generates `TEMPENTRY`

. . . quite complex collection structure

Book Submit New Record page: 1 SUMM

Book

Import data **VDB Relevant** yes no Supported by FZJ-Employee yes no

Institute(s) **R&D Section**

Grant Name Report Number

Author(s) **Title**

Publication Year Language

Conference name Acronym City

Country Begin End

ISBN **Extent**

Edition **Publisher** **Publisher`s City** Series

Volume

Abstract

Additional Information

Please upload your full text Durchsuchen...

Finish & Release **Postpone**

Field designations

Red: mandantory for a full bibliographic description

Black: might not all apply

Blue: save manual work!

Field designations

Red: mandantory for a full bibliographic description

Black: might not all apply

Blue: save manual work!

Import data

…is always available and allows to fetch data from external sources.

Use it if possible!

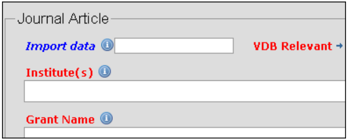
PS: ⓘ is the manual (Wiki-based, german and english)

Import HowTo

- DOI: just insert the doi (e. g. 10.1016/j.physletb.2006.11.038)
- pubmed: just copy from pubmed as displayed (e. g. PMID: 20923669)
- arXiv: just copy from arXiv as displayed (e. g. arxiv:hep-ph/0610431)
- ISBN: use the ISBN-field for this import

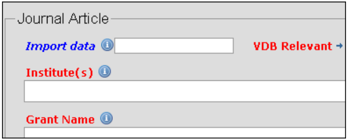
Import HowTo

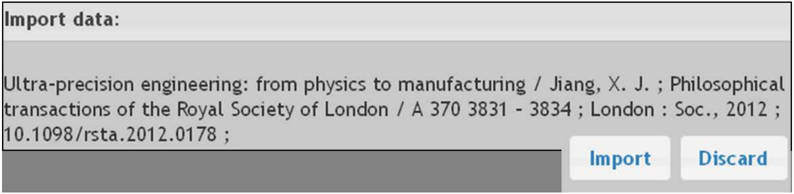
- DOI: just insert the doi (e. g. 10.1016/j.physletb.2006.11.038)
- pubmed: just copy from pubmed as displayed (e. g. PMID: 20923669)
- arXiv: just copy from arXiv as displayed (e. g. arxiv:hep-ph/0610431)
- ISBN: use the ISBN-field for this import



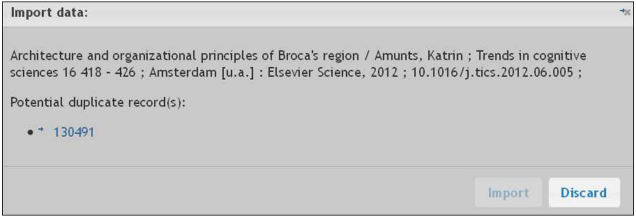
Import HowTo

- DOI: just insert the doi (e. g. 10.1016/j.physletb.2006.11.038)
- pubmed: just copy from pubmed as displayed (e. g. PMID: 20923669)
- arXiv: just copy from arXiv as displayed (e. g. arxiv:hep-ph/0610431)
- ISBN: use the ISBN-field for this import





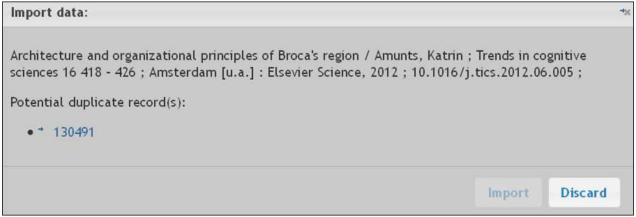
Duplicate entries



At import via doi, pmid, arXiv… JuSER

- can identify potential duplicates
- refuses the import
- shows links to the potential dupes

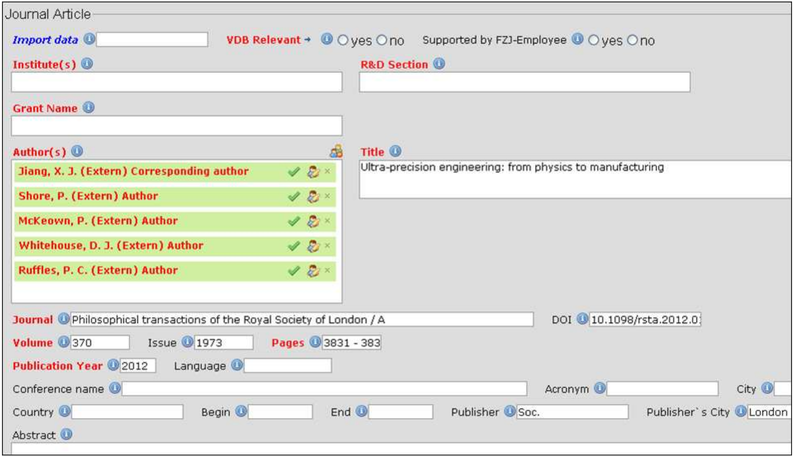
Duplicate entries



At import via doi, pmid, arXiv… JuSER

- can identify potential duplicates
- refuses the import
- shows links to the potential dupes

E. g. DOI Import



Most red fields are filled in already!

Authors

Upon import

authors are a mere guesses by the system. Check them!

Authors

Upon import

authors are a mere guesses by the system. Check them!

- : confirm the guess
- : correct a wrong guess
- : remove an entry
- : clear the whole list

Authors

Upon import

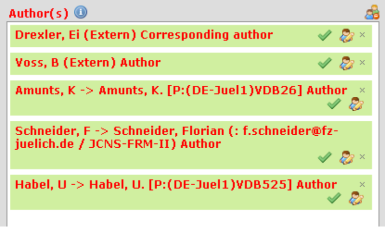
authors are a mere guesses by the system. Check them!

- : confirm the guess
- : correct a wrong guess
- : remove an entry
- : clear the whole list

- Associate only authors that show an email address/institute
- If unsure, leave them red

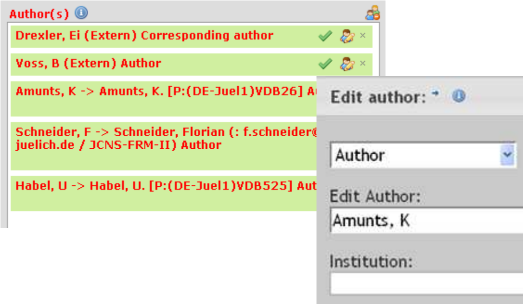
Author input and association

Author input and association



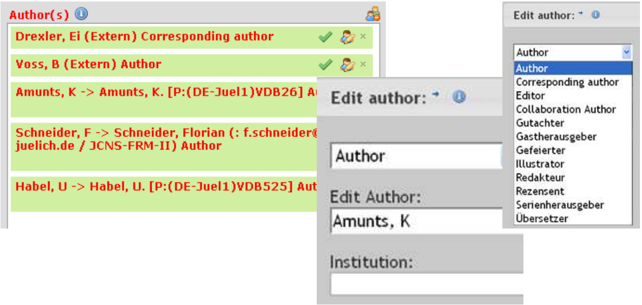
Note: ZB covers publication costs if the *corresponding author* is from Jülich.

Author input and association



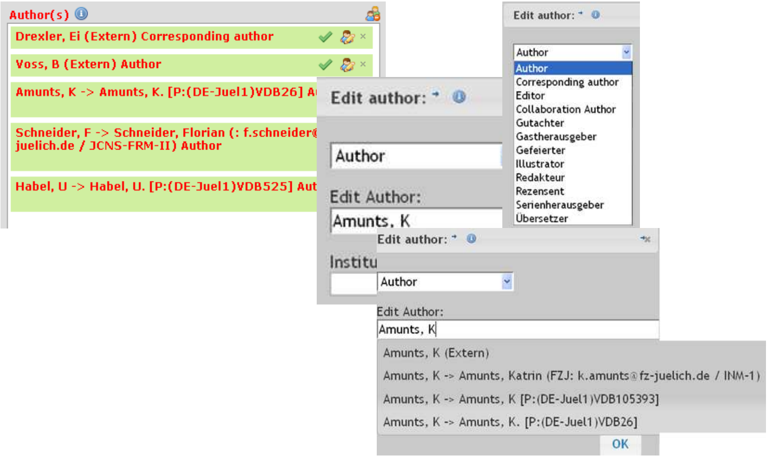
Note: ZB covers publication costs if the *corresponding author* is from Jülich.

Author input and association



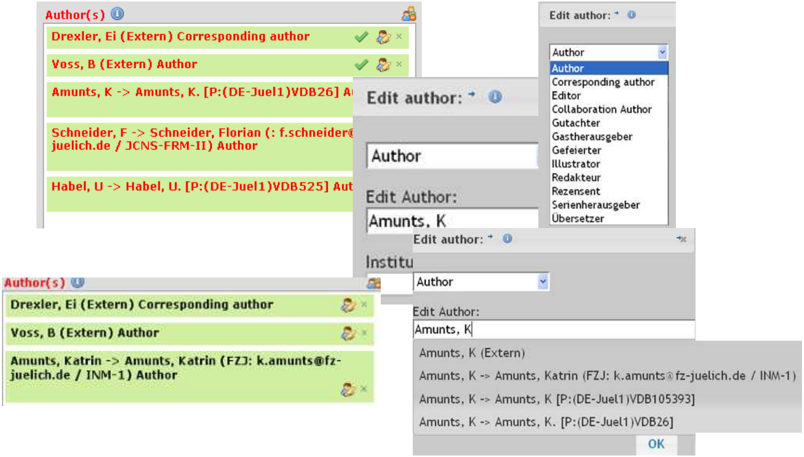
Note: ZB covers publication costs if the *corresponding author* is from Jülich.

Author input and association



Note: ZB covers publication costs if the *corresponding author* is from Jülich.

Author input and association



Note: ZB covers publication costs if the *corresponding author* is from Jülich.

Own publications

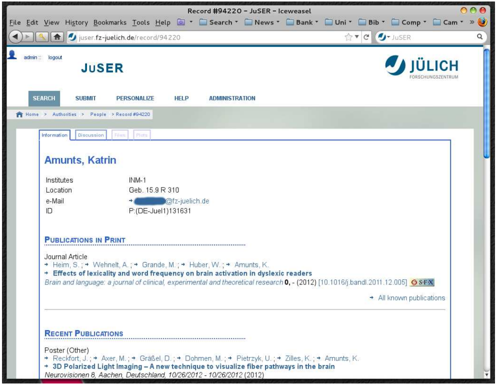
Author association

allows easy extraction of individual and exact publication lists.

Own publications

Author association

allows easy extraction of individual and exact publication lists.



Claim work (required only once!)

1. Log in
2. Navigate to: Authorities / People (http://juser.fz-juelich.de/collection/People)
3. Search for own name (e. g. `'Hofmann, D'`)
4. Note the IDs in question (e. g. `P:(DE-Juel1)VDB63458` and `P:(DE-Juel1)129471`)
5. Open details and check the records found by the link
   `All known publications` (below the list of recent publications)
6. notify `juser@fz-juelich.de`

   (e. g. if the above two people are the same, we need both ids to join them)

Claim work (required only once!)

1. Log in
2. Navigate to: Authorities / People (http://juser.fz-juelich.de/collection/People)
3. Search for own name (e. g. `'Hofmann, D'`)
4. Note the IDs in question (e. g. `P:(DE-Juel1)VDB63458` and `P:(DE-Juel1)129471`)
5. Open details and check the records found by the link `All known publications` (below the list of recent publications)
6. notify `juser@fz-juelich.de`

   (e. g. if the above two people are the same, we need both ids to join them)

To search own publications use `aid:` and own ID in quotes (“”)

e. g. `aid:“P:(DE-Juel1)133794”`

Want only first authorships? Use `fai` instead of `aid`.

JÜLICH FORSCHUNGSZENTRUM

Invenio @ HGF

Part III: Installation

June 9, 2013 | Alexander Wagner

Repositories @ HGF

Roll out and desaster recovery

the same code on at least 5 different instances and keep it consistent by avoiding manual configuration.

Repositories @ HGF

Roll out and desaster recovery

the same code on at least 5 different instances and keep it consistent by avoiding manual configuration.

- git: cds-invenio

Repositories @ HGF

Roll out and desaster recovery

the same code on at least 5 different instances and keep it consistent by avoiding manual configuration.

- git: cds-invenio
  - our version of the main code from CERN
  - used for base system
  - applied by configure/make/make install

Repositories @ HGF

Roll out and desaster recovery

the same code on at least 5 different instances and keep it consistent by avoiding manual configuration.

- git: cds-invenio
  - our version of the main code from CERN
  - used for base system
  - applied by configure/make/make install
- git: hgf-invenio

Repositories @ HGF

Roll out and desaster recovery

the same code on at least 5 different instances and keep it consistent by avoiding manual configuration.

- git: cds-invenio
  - our version of the main code from CERN
  - used for base system
  - applied by configure/make/make install
- git: hgf-invenio
  - overlay for our instances (our additions)
  - our patches (no replacements!)
  - directory structure like /opt/invenio
  - setup routines for roll-out (e. g. collections, rolles, doctypes. . .)
  - instances configs

InstallInvenio

InstallInvenio

sets up the whole instance at each partner in the proper layout including all global (hgf) and local configs

InstallInvenio

InstallInvenio

sets up the whole instance at each partner in the proper layout including all global (hgf) and local configs

- assumes all deps are met and database exists

InstallInvenio

InstallInvenio

sets up the whole instance at each partner in the proper layout including all global (hgf) and local configs

- assumes all deps are met and database exists
- sets up from scratch or from a given point

InstallInvenio

InstallInvenio

sets up the whole instance at each partner in the proper layout including all global (hgf) and local configs

- assumes all deps are met and database exists
- sets up from scratch or from a given point
- compiles and installs INVENIO src

InstallInvenio

InstallInvenio

sets up the whole instance at each partner in the proper layout including all global (hgf) and local configs

- assumes all deps are met and database exists
- sets up from scratch or from a given point
- compiles and installs INVENIO src
- applies hgf patches and configs

InstallInvenio

InstallInvenio

sets up the whole instance at each partner in the proper layout including all global (hgf) and local configs

- assumes all deps are met and database exists
- sets up from scratch or from a given point
- compiles and installs INVENIO src
- applies hgf patches and configs
- applies global and local customizations

InstallInvenio

InstallInvenio

sets up the whole instance at each partner in the proper layout including all global (hgf) and local configs

- assumes all deps are met and database exists
- sets up from scratch or from a given point
- compiles and installs INVENIO src
- applies hgf patches and configs
- applies global and local customizations
- procedures based upon GUI functionality

InstallInvenio

InstallInvenio

sets up the whole instance at each partner in the proper layout including all global (hgf) and local configs

- assumes all deps are met and database exists
- sets up from scratch or from a given point
- compiles and installs INVENIO src
- applies hgf patches and configs
- applies global and local customizations
- procedures based upon GUI functionality
- no need to mouse click in the GUI

InstallInvenio

InstallInvenio

sets up the whole instance at each partner in the proper layout including all global (hgf) and local configs

- assumes all deps are met and database exists
- sets up from scratch or from a given point
- compiles and installs INVENIO src
- applies hgf patches and configs
- applies global and local customizations
- procedures based upon GUI functionality
- no need to mouse click in the GUI
- enables desaster recovery

InstallInvenio

InstallInvenio

sets up the whole instance at each partner in the proper layout including all global (hgf) and local configs

- assumes all deps are met and database exists
- sets up from scratch or from a given point
- compiles and installs INVENIO src
- applies hgf patches and configs
- applies global and local customizations
- procedures based upon GUI functionality
- no need to mouse click in the GUI
- enables desaster recovery
- allows setup of an identical test environment

Mitglied der Helmholtz-Gemeinschaft

JÜLICH
FORSCHUNGSZENTRUM

Invenio @ HGF

Part IV: Backend

June 9, 2013 | Alexander Wagner

HGF “specifics”

Broad areas of research

some INVENIO-HEP-specifics just don’t work

e. g. journal handling, OpenAccess assumption

JÜLICH FORSCHUNGSZENTRUM

HGF “specifics”

Broad areas of research

some INVENIO-HEP-specifics just don’t work

e. g. journal handling, OpenAccess assumption

- Journal handling based on IDs, not names

HGF “specifics”

Broad areas of research

some INVENIO-HEP-specifics just don’t work

e. g. journal handling, OpenAccess assumption

- Journal handling based on IDs, not names
- allow for foreign data ingests (e. g. ebook packages)

HGF "specifics"

Broad areas of research

some INVENIO-HEP-specifics just don't work

e. g. journal handling, OpenAccess assumption

- Journal handling based on IDs, not names
- allow for foreign data ingests (e. g. ebook packages)
  - important in context of a libraries catalogue
  - stick to MARC definitions (Library of Congress)
  - avoid specialities (exceptions: 9xx)
  - implies adoption even of default defs (e. g. internal Marc, BibTEX ⌒)

JÜLICH
FORSCHUNGSZENTRUM

HGF “specifics”

Broad areas of research

some INVENIO-HEP-specifics just don’t work

e. g. journal handling, OpenAccess assumption

- Journal handling based on IDs, not names
- allow for foreign data ingests (e. g. ebook packages)
  - important in context of a libraries catalogue
  - stick to MARC definitions (Library of Congress)
  - avoid specialities (exceptions: 9xx)
  - implies adoption even of default defs (e. g. internal Marc, BibTEX⌢)
- Normalize all you can get

HGF “specifics”

Broad areas of research

some INVENIO-HEP-specifics just don’t work

e. g. journal handling, OpenAccess assumption

- Journal handling based on IDs, not names
- allow for foreign data ingests (e. g. ebook packages)
  - important in context of a libraries catalogue
  - stick to MARC definitions (Library of Congress)
  - avoid specialities (exceptions: 9xx)
  - implies adoption even of default defs (e. g. internal Marc, BibTEX ⌢)
- Normalize all you can get
  - to be handled by librarians, not programmers
  - vitally important: Authorities

Modules

- Importer: (crossref, pubmed, GVK, arxiv, inspire. . .)

Modules

- Importer: (crossref, pubmed, GVK, arxiv, inspire. . .)
  - hook up with web services
  - reuse old code bascially in perl (to be rewritten to Python as time allows)
  - add new stuff in Python

Modules

- Importer: (crossref, pubmed, GVK, arxiv, inspire. . .)
  - hook up with web services
  - reuse old code bascially in perl (to be rewritten to Python as time allows)
  - add new stuff in Python
- HGFImport.py (call externals but allow for permission checking)

Modules

- Importer: (crossref, pubmed, GVK, arxiv, inspire. . .)
  - hook up with web services
  - reuse old code bascially in perl (to be rewritten to Python as time allows)
  - add new stuff in Python
- HGFImport.py (call externals but allow for permission checking)
- HandleNames.py (author guessing)

Modules

- Importer: (crossref, pubmed, GVK, arxiv, inspire. . .)
  - hook up with web services
  - reuse old code bascially in perl (to be rewritten to Python as time allows)
  - add new stuff in Python
- HGFImport.py (call externals but allow for permission checking)
- HandleNames.py (author guessing)
- JSGetAllChildren.py (recurse to the end of tree like MARC structures)

Modules

- Importer: (crossref, pubmed, GVK, arxiv, inspire. . .)
  - hook up with web services
  - reuse old code bascially in perl (to be rewritten to Python as time allows)
  - add new stuff in Python
- HGFImport.py (call externals but allow for permission checking)
- HandleNames.py (author guessing)
- JSGetAllChildren.py (recurse to the end of tree like MARC structures)
- PubExporter.py (handle specifics needed for web export)

Modules

- Importer: (crossref, pubmed, GVK, arxiv, inspire. . .)
  - hook up with web services
  - reuse old code bascially in perl (to be rewritten to Python as time allows)
  - add new stuff in Python
- HGFImport.py (call externals but allow for permission checking)
- HandleNames.py (author guessing)
- JSGetAllChildren.py (recurse to the end of tree like MARC structures)
- PubExporter.py (handle specifics needed for web export)

Other functions

Always try to use INVENIO functions like alerts, baskets, . . .

Modules

- Importer: (crossref, pubmed, GVK, arxiv, inspire. . .)
  - hook up with web services
  - reuse old code bascially in perl (to be rewritten to Python as time allows)
  - add new stuff in Python
- HGFImport.py (call externals but allow for permission checking)
- HandleNames.py (author guessing)
- JSGetAllChildren.py (recurse to the end of tree like MARC structures)
- PubExporter.py (handle specifics needed for web export)

Other functions

Always try to use INVENIO functions like alerts, baskets, . . .

Every unwritten line of code is a good line of code!

Types of records

- People e. g. authors

Types of records

- People e. g. authors
  - individualize(!) local authors
  - entries are real people, not names (distinguish Wagner, A. and Wagner, A.)
  - connect to local institutes (incl. history)
  - allow interchange of records (e. g. Jülich with RWTH Aachen and MLZ)

Types of records

- People e. g. authors
  - individualize(!) local authors
  - entries are real people, not names (distinguish Wagner, A. and Wagner, A.)
  - connect to local institutes (incl. history)
  - allow interchange of records (e. g. Jülich with RWTH Aachen and MLZ)
- Institutes (people-like aggregates)

Types of records

- People e. g. authors
  - individualize(!) local authors
  - entries are real people, not names (distinguish Wagner, A. and Wagner, A.)
  - connect to local institutes (incl. history)
  - allow interchange of records (e. g. Jülich with RWTH Aachen and MLZ)
- Institutes (people-like aggregates)
  - keep track of history (renaming, merging, splitting)
  - horizonal and vertical linkage
  - complex topography in names (e. g. loops)

Types of records

- People e. g. authors
  - individualize(!) local authors
  - entries are real people, not names (distinguish Wagner, A. and Wagner, A.)
  - connect to local institutes (incl. history)
  - allow interchange of records (e. g. Jülich with RWTH Aachen and MLZ)
- Institutes (people-like aggregates)
  - keep track of history (renaming, merging, splitting)
  - horizonal and vertical linkage
  - complex topography in names (e. g. loops)
- Grants (money-like aggregats)

Types of records

- People e. g. authors
  - individualize(!) local authors
  - entries are real people, not names (distinguish Wagner, A. and Wagner, A.)
  - connect to local institutes (incl. history)
  - allow interchange of records (e. g. Jülich with RWTH Aachen and MLZ)
- Institutes (people-like aggregates)
  - keep track of history (renaming, merging, splitting)
  - horizonal and vertical linkage
  - complex topography in names (e. g. loops)
- Grants (money-like aggregats)
  - horizoncal as well as vertical connections
  - different types (POF (HGF-specific), EU, DFG, . . .)
  - flexibly extensible by librarians
  - visibility in websubmit depending on instance

Types of records

- Periodicals, Journals

Types of records

- Periodicals, Journals
  - $\approx$ 30.000 journals (links to german ZDB and EZB)
  - complex external tool to build up records
  - autogenerated/updated once a year (trigger: new JCR)
  - allow harvesting of records generated
  - contain statistics keys for evaluation
  - ISSN does not work as key: need to connect print and online edition to one journal

Types of records

- Periodicals, Journals
  - $\approx$ 30.000 journals (links to german ZDB and EZB)
  - complex external tool to build up records
  - autogenerated/updated once a year (trigger: new JCR)
  - allow harvesting of records generated
  - contain statistics keys for evaluation
  - ISSN does not work as key: need to connect print and online edition to one journal
- Statistics keys

Types of records

- Periodicals, Journals
  - $\approx$ 30.000 journals (links to german ZDB and EZB)
  - complex external tool to build up records
  - autogenerated/updated once a year (trigger: new JCR)
  - allow harvesting of records generated
  - contain statistics keys for evaluation
  - ISSN does not work as key: need to connect print and online edition to one journal
- Statistics keys
  - database coverage (journal listed in Scopus or WoS, JCR or Pubmed. . .)
  - OpenAccess designation
  - license identifier (Allianz-licensing)

Types of records

- Periodicals, Journals
  - $\approx$ 30.000 journals (links to german ZDB and EZB)
  - complex external tool to build up records
  - autogenerated/updated once a year (trigger: new JCR)
  - allow harvesting of records generated
  - contain statistics keys for evaluation
  - ISSN does not work as key: need to connect print and online edition to one journal
- Statistics keys
  - database coverage (journal listed in Scopus or WoS, JCR or Pubmed. . .)
  - OpenAccess designation
  - license identifier (Allianz-licensing)
- Vocabulary

Types of records

- Periodicals, Journals
  - $\approx$ 30.000 journals (links to german ZDB and EZB)
  - complex external tool to build up records
  - autogenerated/updated once a year (trigger: new JCR)
  - allow harvesting of records generated
  - contain statistics keys for evaluation
  - ISSN does not work as key: need to connect print and online edition to one journal
- Statistics keys
  - database coverage (journal listed in Scopus or WoS, JCR or Pubmed. . .)
  - OpenAccess designation
  - license identifier (Allianz-licensing)
- Vocabulary
  - allow for classifications (e. g. POF level)

Frontend

- rich, browserbased GUI: AJAX and CSS

Frontend

- rich, browserbased GUI: AJAX and CSS
- JavaScript: JQuery + x

Frontend

- rich, browserbased GUI: AJAX and CSS
- JavaScript: JQuery + x
- use few different elements (mainly tokeninput, autocomplete, datepicker)

Frontend

- rich, browserbased GUI: AJAX and CSS
- JavaScript: JQuery + x
- use few different elements (mainly tokeninput, autocomplete, datepicker)
- browser independent (needs to cover at least IE8+, Safari, FireFox on Win, Mac and Linux)

Frontend

- rich, browserbased GUI: AJAX and CSS
- JavaScript: JQuery + x
- use few different elements (mainly tokeninput, autocomplete, datepicker)
- browser independent (needs to cover at least IE8+, Safari, FireFox on Win, Mac and Linux)
- Tokeninput (heavily extended and debugged(!) version of jquery.tokeninput)

Frontend

- rich, browserbased GUI: AJAX and CSS
- JavaScript: JQuery + x
- use few different elements (mainly tokeninput, autocomplete, datepicker)
- browser independent (needs to cover at least IE8+, Safari, FireFox on Win, Mac and Linux)
- Tokeninput (heavily extended and debugged(!) version of jquery.tokeninput)
  - institutes
  - authors
  - grants

Frontend

- rich, browserbased GUI: AJAX and CSS
- JavaScript: JQuery + x
- use few different elements (mainly tokeninput, autocomplete, datepicker)
- browser independent (needs to cover at least IE8+, Safari, FireFox on Win, Mac and Linux)
- Tokeninput (heavily extended and debugged(!) version of jquery.tokeninput)
  - institutes
  - authors
  - grants
- JSON-returns from the backend

Frontend

- rich, browserbased GUI: AJAX and CSS
- JavaScript: JQuery + x
- use few different elements (mainly tokeninput, autocomplete, datepicker)
- browser independent (needs to cover at least IE8+, Safari, FireFox on Win, Mac and Linux)
- Tokeninput (heavily extended and debugged(!) version of jquery.tokeninput)
  - institutes
  - authors
  - grants
- JSON-returns from the backend
- avoid program logic (backend has to deliver ready to use data)

Frontend

- rich, browserbased GUI: AJAX and CSS
- JavaScript: JQuery + x
- use few different elements (mainly tokeninput, autocomplete, datepicker)
- browser independent (needs to cover at least IE8+, Safari, FireFox on Win, Mac and Linux)
- Tokeninput (heavily extended and debugged(!) version of jquery.tokeninput)
  - institutes
  - authors
  - grants
- JSON-returns from the backend
- avoid program logic (backend has to deliver ready to use data)
- use encoding scheme for returns (allow processing in loops)

Frontend

- rich, browserbased GUI: AJAX and CSS
- JavaScript: JQuery + x
- use few different elements (mainly tokeninput, autocomplete, datepicker)
- browser independent (needs to cover at least IE8+, Safari, FireFox on Win, Mac and Linux)
- Tokeninput (heavily extended and debugged(!) version of jquery.tokeninput)
  - institutes
  - authors
  - grants
- JSON-returns from the backend
- avoid program logic (backend has to deliver ready to use data)
- use encoding scheme for returns (allow processing in loops)
- define hidden subfields to keep structures (e. g. repeatable fields)

Frontend

- rich, browserbased GUI: AJAX and CSS
- JavaScript: JQuery + x
- use few different elements (mainly tokeninput, autocomplete, datepicker)
- browser independent (needs to cover at least IE8+, Safari, FireFox on Win, Mac and Linux)
- Tokeninput (heavily extended and debugged(!) version of jquery.tokeninput)
  - institutes
  - authors
  - grants
- JSON-returns from the backend
- avoid program logic (backend has to deliver ready to use data)
- use encoding scheme for returns (allow processing in loops)
- define hidden subfields to keep structures (e. g. repeatable fields)
- invisibly enrich data from the backend (e. g. IDs, DDC, statistics keys. . .)

Backend

Expose records in JSON structures to JavaScript

Backend

Expose records in JSON structures to JavaScript

- no repeatable subfields simplifies JSON

Backend

Expose records in JSON structures to JavaScript

- no repeatable subfields simplifies JSON
- define new output format: JS

Backend

Expose records in JSON structures to JavaScript

- no repeatable subfields simplifies JSON
- define new output format: JS
- define format templates for each record type

Backend

Expose records in JSON structures to JavaScript

- no repeatable subfields simplifies JSON
- define new output format: JS
- define format templates for each record type
  - use webbased frontend if possible
  - use pythonic BFE only for complex stuff
  - complex returns as ready to use text (i. e. escape substructures to strings)

Backend

Expose records in JSON structures to JavaScript

- no repeatable subfields simplifies JSON
- define new output format: JS
- define format templates for each record type
  - use webbased frontend if possible
  - use pythonic BFE only for complex stuff
  - complex returns as ready to use text (i. e. escape substructures to strings)
- return hashes keyed as MARC:

Backend

Expose records in JSON structures to JavaScript

- no repeatable subfields simplifies JSON
- define new output format: JS
- define format templates for each record type
  - use webbased frontend if possible
  - use pythonic BFE only for complex stuff
  - complex returns as ready to use text (i. e. escape substructures to strings)
- return hashes keyed as MARC:
  1. Field/Subfield: `I245__a`
  2. Structure: `I536__`

Backend

Expose records in JSON structures to JavaScript

- no repeatable subfields simplifies JSON
- define new output format: JS
- define format templates for each record type
  - use webbased frontend if possible
  - use pythonic BFE only for complex stuff
  - complex returns as ready to use text (i. e. escape substructures to strings)
- return hashes keyed as MARC:
  1. Field/Subfield: `I245__a`
  2. Structure: `I536__`
- innerrecordlinks: return from the backend (e. g. `536__` and `913__`)

Data sources

- Local phone book (LDAP)

Data sources

- Local phone book (LDAP)
- LDAP harvesting (once a night, currently cron)

Data sources

- Local phone book (LDAP)
- LDAP harvesting (once a night, currently cron)
- unique IDs over time (email not suitable: limited TTL, recycling for common names)

Data sources

- Local phone book (LDAP)
- LDAP harvesting (once a night, currently cron)
- unique IDs over time (email not suitable: limited TTL, recycling for common names)
- allow for arbitrary IDs

Data sources

- Local phone book (LDAP)
- LDAP harvesting (once a night, currently cron)
- unique IDs over time (email not suitable: limited TTL, recycling for common names)
- allow for arbitrary IDs
- allow for multiple IDs per entity (0247_ vs. 035__a)

Data sources

- Local phone book (LDAP)
- LDAP harvesting (once a night, currently cron)
- unique IDs over time (email not suitable: limited TTL, recycling for common names)
- allow for arbitrary IDs
- allow for multiple IDs per entity (0247_ vs. 035__a)
- allow usage of IDs from several sources

Data sources

- Local phone book (LDAP)
- LDAP harvesting (once a night, currently cron)
- unique IDs over time (email not suitable: limited TTL, recycling for common names)
- allow for arbitrary IDs
- allow for multiple IDs per entity (0247_ vs. 035__a)
- allow usage of IDs from several sources

Data sources

- Local phone book (LDAP)
- LDAP harvesting (once a night, currently cron)
- unique IDs over time (email not suitable: limited TTL, recycling for common names)
- allow for arbitrary IDs
- allow for multiple IDs per entity (0247_ vs. 035__a)
- allow usage of IDs from several sources

Goal

Implement ORCiD linkup with automagic registration/fetching

Statistics

JuSER was brought to you by

Deutsches Elektronensynchrotron, Zentralbibliothek

Forschungszentrum Jülich, Zentralbibliothek

GSI Helmholtzzentrum für Schwerionenforschung, Biblitothek + Kern-IT

RWTH Aachen, Hochschulbibliothek

Maier-Leibnitz-Zentrum, Garching

JuSER is powered by INVENIO developed at CERN

Questions?

Alexander Wagner
Zentralbibliothek

Fachinformation,
wiss. Publizieren

Tel.: 1586
a.wagner@fz-juelich.de

http://www.fz-juelich.de/zb

This document is available as FZJ-2013-499

Typeset by pdfLaTeX