

From strings to things

A Linked Open Data API for library hackers and web developers

SWIB 2013, Hamburg

November 27th, 2013



Linked Open Data

- Interoperability through common, flexible data model and common identifiers
- `<Typee> <was written by> <Melville>`
- `<http://lobid.org/resource/HT002189125>`
`<http://purl.org/dc/elements/1.1/creator>`
`<http://d-nb.info/gnd/118580604> .`

Message

- So our message has been: Use things, not strings!
- e.g.
`http://d-nb.info/gnd/118580604`,
not 'Melville, Herman', 'Herman Melville', 'H. Melville', etc.
- But: where to get these IDs from?



CC-SA-2.0 Infrogation of New Orleans,
Wikimedia Commons,
File:WrongWayCarrolltonNOLA.JPG

Message

“Clothes are great, so please learn knitting”



CC-BY-2.0 Angela Montillon, Wikimedia Commons, File:Colourful_wool_2.jpg
CC-SA-2.5 Wikimedia Commons, File:Knit4.jpg
CC-BY-SA-3.0 Jomegat, Wikimedia Commons, File:Knitting_dropped_stitch_5.jpg

Response

- “OK, but can’t I just wear some clothes? Do I have to create them myself, manually?”
- Do you have to be a LOD expert to benefit from LOD?



CC-BY-2.0 Andrew Vargas, Wikimedia Commons, File:Well-clothed_baby.jpg

lobid.org

- lobid.org: LOD service of hbz, since 2010
- title data of union catalog (lobid-resources), authority data (lobid-organisations)
- Dumps, resolvable URIs, content negotiation, RDFa, SPARQL (triple store)
- different problems, new requirements → developed a new backend since late 2012

Problems

- General performance issues: complex queries causing triple store hang ups
- Specific performance-critical use cases: auto suggest, e.g. for authority data
- Technological obscurity: Semantic Web, cutting edge since 2001. Our goal: provide data, not just evangelize technology

Approach

- Fix performance problems: stabilize current applications and enable new use cases
- Put the web and web developers into focus
- LOD for web devs, not only for LOD experts

Approach

JSON over HTTP

API: what

- Application programming interfaces: essential for reusable software modules
- These modules communicate only via their API, they know no implementation details
- So implementations become exchangeable – without requiring changes in API clients

API: why

- Only with a stable API, modules are *actually* reusable: reuse has to work
- Triple store or search index not suitable as an API: should provide a stable abstraction over implementation details and the data

API: requests

GET /resource?id=0940450003

GET /resource?name=Typee

GET /organisation?id=DE-605

GET /organisation?name=hbz

GET /person?id=118580604

GET /person?name=Herman+Melville

API: responses

```
GET /person?name=Ernest+Hem&format=short
```

```
[  
"Hemingway, Ernest (1899-1961)",  
"Hemann, Augustin Ernst Roman (1748-1820)",  
"Hempel, Ernst Wilhelm (1745-1799)",  
"Jamaigne, Jean Ernest de",  
"Lacheman, Ernest R. (1906-1982)",  
"Uthemann, Ernest W. (1953-)"  
]
```

API: usage

This can be used for an auto suggest feature:

Ernest Hem | Search

- Hemingway, Ernest (1899-1961)
- Hemmann, Augustin Ernst Roman (1748-1820)
- Hempel, Ernst Wilhelm (1745-1799)
- Jamaigne, Jean Ernest de
- Lacheman, Ernest R. (1906-1982)
- Uthemann, Ernest W. (1953-)
- Wirzén, Johan Ernst Adhemar (1812-1857)

ids parameter to get the suggestions.
e inserted value. See the implementation
e source of this page.
ita from a remote URL (i.e. a different
ng JSONP (use full URL in your code, i.e.
ute).

When a suggestion is selected, insert its ID:

http://d-nb.info/gnd/118549030 | Search

API: from strings to things

That actually uses a different response format:

```
GET http://api.lobid.org/person?name=Ernest+Hem&format=ids
```

```
[{
  label: "Hemingway, Ernest (1899-1961)",
  value: "http://d-nb.info/gnd/118549030"
},{
  label: "Hemmann, Augustin Ernst Roman (1748-1820)",
  value: "http://d-nb.info/gnd/130030252"
},{
  label: "Hempel, Ernst Wilhelm (1745-1799)",
  value: "http://d-nb.info/gnd/100292437"
}]
```

API: from strings to things

```
GET http://api.lobid.org/person?id=118549030&format=full
```

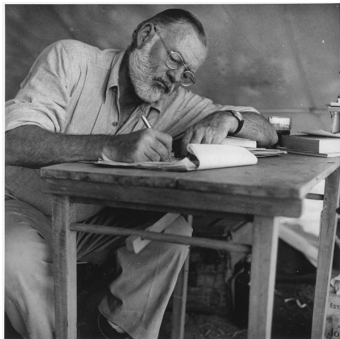
```
[{
  @id: "http://d-nb.info/gnd/118549030",
  preferredNameForThePerson: "Hemingway, Ernest",
  dateOfBirth: "1899",
  dateOfDeath: "1961",
  variantNameForThePerson: [
    "Heminguej, E.", ...
  ],
  placeOfBirth: "http://d-nb.info/gnd/4461931-5",
  sameAs: "http://dbpedia.org/resource/Ernest_Hemingway",
  wikipedia: "http://de.wikipedia.org/wiki/Ernest_Hemingway",
  ...
  @context: "http://api.lobid.org/context/gnd.json"
}]
```


API: from strings to things

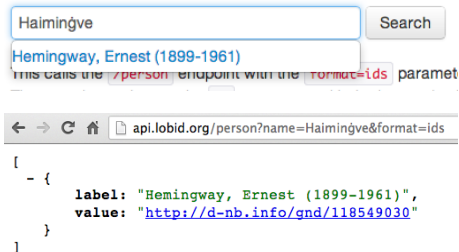
All alternative names:

```
variantNameForThePerson: [  
  "Heminguej, E.",  
  "Hemingye, Ernest",  
  "Cheminhuej, Ernest",  
  "Heminguei, E.",  
  "KheminguéI, Ernest",  
  "Cheminguéj, Ernest",  
  "Hai-ming-wei, ...",  
  "Haimingve",  
  "Hemingvejs, Ernests",  
  "Hemingway, Ernest Miller",  
  "Heminguei, Ernest",  
  "Hemingvej, Ernest",  
  "Hamingwáj, Arnist",  
  "Haminghwáj, Arnist",  
  "Himingwáj",  
  "Himingwáj, Arnist",  
  "Hemingwei, ...",  
  "Hemingway, E.",  
  "Himinghwáj, Arnist",  
  "Hemingway",  
  "Hayminghwáj, Arnist",  
  "Hemingyey, Ernest",  
  "Hamingwáj, Arnist",  
  "Cheminguaiš, Ernest",  
  "Hemingway, Ernest M.",  
  "海明威"]
```

For: <http://d-nb.info/gnd/118549030>



API: from strings to things



The screenshot shows a search interface with a text input containing "Haimingve" and a "Search" button. Below the input, a dropdown menu displays "Hemingway, Ernest (1899-1961)". A red box highlights the text "this calls the /person endpoint with the format=ids parameter". Below this, a browser address bar shows the URL "api.lobid.org/person?name=Haimingve&format=ids". The response is a JSON object:

```
{
  - {
    label: "Hemingway, Ernest (1899-1961)",
    value: "http://d-nb.info/gnd/118549030"
  }
}
```

LOD and Semantic Web technology enable that.
But we shouldn't expect anyone to learn RDF,
SPARQL, etc for such a simple use case

API: but where's the LOD

- “But where are the unified IDs in the keys of the JSON response? It's just strings!”
- Enter JSON-LD: @context maps plain JSON keys to URIs → API as abstraction
- JSON-LD also enables RDF serialization, available from API via content negotiation

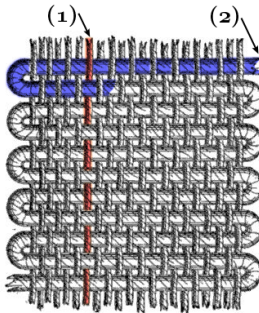
API: documentation

Sample queries, documentation on parameters and content negotiation, auto suggest samples with Javascript code, etc:

`http://api.lobid.org/`

Technology

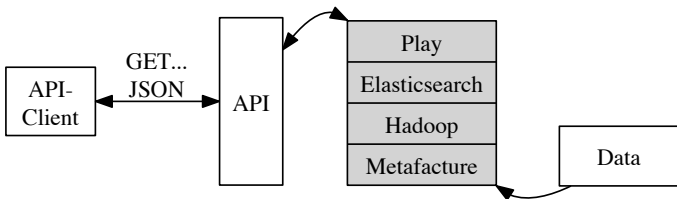
Community needs to build and share know-how:



CC-BY-2.0 Angela Montillon, Wikimedia Commons, File:Colourful_wool_2.jpg
CC-BY-SA-3.0 Rjy, derivative: Derwok, Wikimedia Commons, File:Kette_und_Schub_num_col.jpg
CC-BY-2.0 Tony Hsgett, Wikimedia Commons, File:Coloured_cloth_2_(3539454254).jpg

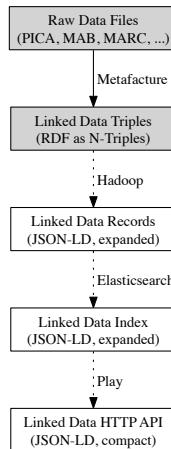
Technology

Our technology stack:
Metafactory, Hadoop, Elasticsearch, Play



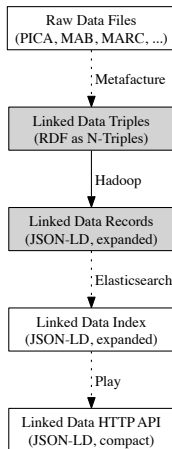
Technology

- Raw data to N-Triples: Metafactory
- N-Triples to JSON-LD records: Hadoop
- Indexing JSON-LD: Elasticsearch
- HTTP API: Play-Framework



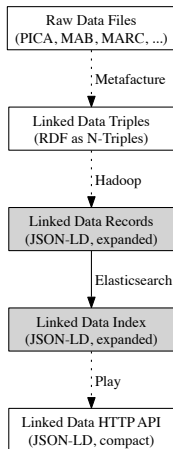
Technology

- Raw data to N-Triples: Metafactory
- N-Triples to JSON-LD records: Hadoop
- Indexing JSON-LD: Elasticsearch
- HTTP API: Play-Framework



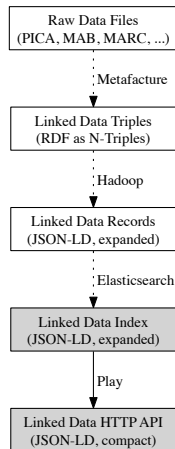
Technology

- Raw data to N-Triples: Metafactory
- N-Triples to JSON-LD records: Hadoop
- **Indexing JSON-LD: Elasticsearch**
- HTTP API: Play-Framework



Technology

- Raw data to N-Triples: Metafactory
- N-Triples to JSON-LD records: Hadoop
- Indexing JSON-LD: Elasticsearch
- HTTP API: Play-Framework



Metafactory: tools

A tool suite for metadata processing

<https://github.com/culturegraph/metafactory-core/wiki>

<https://github.com/culturegraph/metafactory-ide/wiki>

Metafactory - metadata-transformation/sample.flux - Eclipse SDK - /Users/fsteeg/workspace

```
1 default: files = FLUX_DIR;
2 files + "input.xml" |
3 open-file |
4 decode-xml |
5 handle-marcxml |
6 morph(files + "morph.xml") |
7 stream-tee |
8   encode-ntriples |
9   write(files + "output.nt")
10 } |
11 encode-dot |
12 write(files + "output.dot")
13 }
14 handle-generic-xml
15   handle-marcxml
16   handle-picaxml
17   jsript
18   log-object
19   log-stream
20   log-time
21   match
22   merge-batch-stream
23   morph
24   normalize-utf8
```

Description: Benchmarks the execution time of the downstream modules.

In: class java.lang.Object

Outs: class java.lang.Object

Implementation:
org.culturegraph.mf.stream.pipe.ObjectTimer

Hadoop: configuration

Config of properties for JSON-LD records:

```
resolve = . \ ¶  
» http://purl.org/dc/elements/1.1/creator; \ ¶  
» http://purl.org/dc/elements/1.1/contributor; \ ¶  
» http://purl.org/dc/terms/subject; \ ¶  
» http://www.w3.org/2003/01/geo/wgs84_pos#location; \ ¶  
» http://www.w3.org/2006/vcard/ns#adr; \ ¶  
» http://purl.org/lobid/lv#fundertype; \ ¶  
» http://purl.org/lobid/lv#stocksize ¶  
¶  
predicates = . \ ¶  
» http://www.w3.org/1999/02/22-rdf-syntax-ns#type; \ ¶  
» http://d-nb.info/standards/elementset/gnd#preferredNameForThePerson; \ ¶  
» http://d-nb.info/standards/elementset/gnd#dateOfBirth; \ ¶  
» http://d-nb.info/standards/elementset/gnd#dateOfDeath; \ ¶  
» http://www.w3.org/2004/02/skos/core#prefLabel; \ ¶  
» http://www.w3.org/2003/01/geo/wgs84_pos#lat; \ ¶  
» http://www.w3.org/2003/01/geo/wgs84_pos#long; \ ¶
```

Elasticsearch: indexes

Index overview in Elasticsearch-Head-Plugin:

**lobid-item-
index-
20131008-
105029**

size: 14.7gb
(14.7gb)
docs: 65561846
(65561846)

Info ▾

Actions ▾

**lobid-item-
index-
20131027-
054907**

size: 14.7gb
(14.7gb)
docs: 65568116
(65568116)

Info ▾

Actions ▾

**lobid-item-
index-
20131103-
091829**

size: 14.7gb
(14.7gb)
docs: 65568116
(65568116)

Info ▾

Actions ▾

**lobid-orgs-
index-
20131031-
143552**

size: 62.7mb
(62.7mb)
docs: 43955
(43955)

Info ▾

Actions ▾

**lobid-orgs-
index-
20131104-
094303**

size: 62.6mb
(62.6mb)
docs: 43957
(43957)

Info ▾

Actions ▾

lobid-items X

lobid-
organisations X

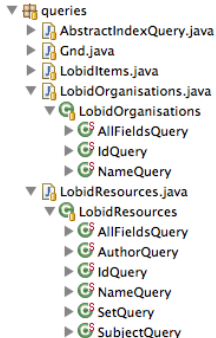
lobid-
organisations-
staging X

lobid-items-
staging X

Play: queries

Elasticsearch queries from Play controllers:

```
» public static class NameQuery extends AbstractIndexQuery {  
    |  
    | @Override |  
    | public List<String> fields() {  
    | » » return Arrays.asList(  
    | » » » "graph:http://xmlns.com/foaf/0.1/name.@value",  
    | » » » "graph:http://www.w3.org/2004/02/skos/core#prefLabel.@value");  
    | » » }  
    | |  
    | @Override |  
    | public QueryBuilder build(final String queryString) {  
    | » » return multiMatchQuery(queryString, fields().toArray(new String[] {}));  
    | » » » .operator(Operator.AND);  
    | » » }  
    | }  
}
```



Technology: documentation

Details on how this works, the actual code and workflows, collaboration infrastructure, etc:

<http://github.com/lobid/lodmill/>

Operations: overview



- Apache as proxy for continuous operation
- Play API server shared with Elasticsearch
- Elasticsearch: 3 servers, 1 productive
- Hadoop: 5 servers, configured with Puppet

Operations: overview



- Apache as proxy for continuous operation
- Play API server shared with Elasticsearch
- Elasticsearch: 3 servers, 1 productive
- Hadoop: 5 servers, configured with Puppet

Operations: overview



- Apache as proxy for continuous operation
- Play API server shared with Elasticsearch
- **Elasticsearch: 3 servers, 1 productive**
- Hadoop: 5 servers, configured with Puppet

Operations: overview



- Apache as proxy for continuous operation
- Play API server shared with Elasticsearch
- Elasticsearch: 3 servers, 1 productive
- Hadoop: 5 servers, configured with Puppet



Operations: what we like

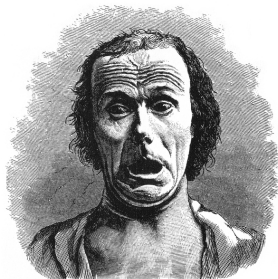
- Technology stack: config of transformations, queries, views
- JSON-LD, @context
- Data updates without affecting production
- Elasticsearch performance



CC-0, Wikimedia Commons,
File:Expression_of_the_Emotions_Figure_17.png

Operations: what we don't like

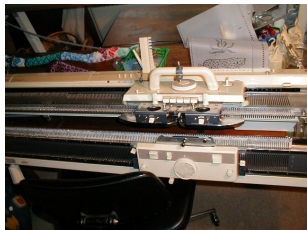
- Manual deployment, proxy and index switching
- Long feedback cycle for full transformation
- Goal: automation and faster indexing



CC-0, Wikimedia Commons,
File:Expression_of_the_Emotions_Figure_20.png

Operations: summary

So not completely there yet, still some manual work involved, but much more than just the yarn



CC-BY-2.0 Angela Montillon, Wikimedia Commons, File:Colourful_wool_2.jpg
CC-SA-3.0 Gudde Fog, Wikimedia Commons, File:MachineKnittingKnittax.jpg
CC-SA-2.0 Joep anker, Wikimedia Commons, File:WLANL_-_jpa2003_-_knit_and_wear_v1akbreimachine (2007) .jpg

Usage

- For progress, usage and feedback is key
- Internal users: e.g. lobid.org, repository cataloging, regional bibliography in 2014
- External users: in contact with various libraries and related institutions

Feedback

- Had early internal reviews, early external beta, got important feedback
- Feedback & iteration crucial: can't guess what's useful, have to find out with users



CC-SA-2.0 lumaxart, Wikimedia Commons.
File:Working_Together_Teamwork_Puzzle_Concept.jpg

Openness

- Code, but also processes open: issues, CI, code reviews, wiki on GitHub
<http://github.com/lobid/>
- Open API:
<http://api.lobid.org/>
- We're very happy about usage, feedback, contributions on all levels



CC BY-NC-SA 2.0, JohnEdgarPark,
<http://www.flickr.com/photos/edgar/2951139311/>

Contact

steeg@hbz-nrw.de, @fsteeeg
christoph@hbz-nrw.de, @dr0ide

These slides are licensed under CC BY-NC-SA 3.0 as required by material used
<http://creativecommons.org/licenses/by-nc-sa/3.0/>

