

MARKLOGIC

NOSQL DATABASE FOR SCIENTIFIC PUBLICATIONS

Feasibility Study

Bo Werth, STI/EAS

PUBLICATION DATABASES

SPRINGER

link.springer.com

ELSEVIER

sciencedirect.com

[Scopus](https://scopus.com)

ELSEVIER SCOPUS

- abstract and citation database of peer-reviewed literature
 - scientific journals, books and conference proceedings
- comprehensive overview of the world's research output
 - science & technology, medicine, social sciences, arts & humanities
- smart tools to track, analyze and visualize research

SCOPUS CUSTOM DATA

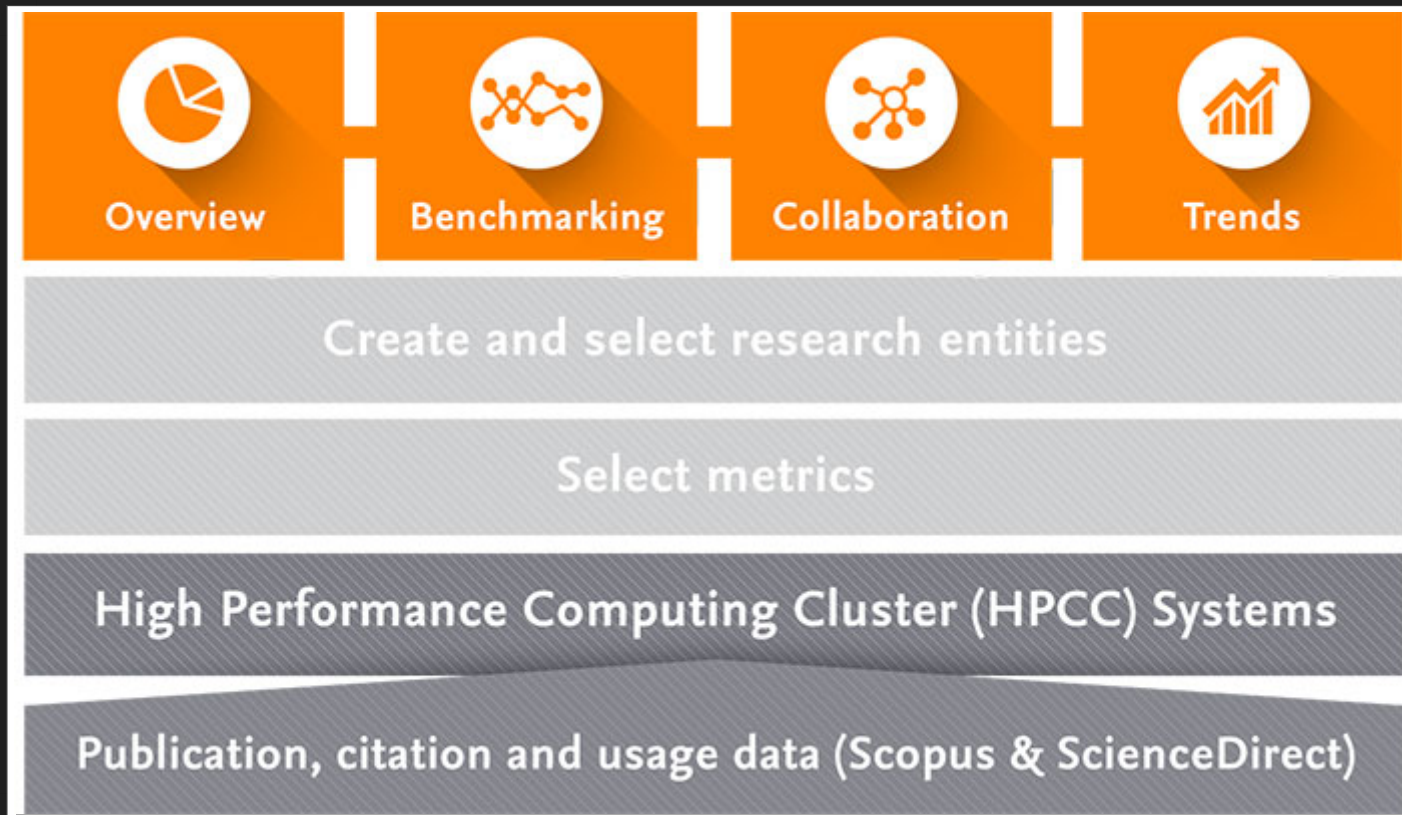
- Time: 1996-2013
- Size: ~200GB per year,
total: more than 1.3 GB decompressed

DATA FORMAT

- batches of 10.000 pgp encrypted, compressed archives
 - separate XML files for article and reverse citations
 - article file contains approx. 450 fields

INDICATOR DERIVATION

Elsevier **SciVal**



MARKLOGIC

- "Any Structure", schema-agnostic, no pre-defined model
 - Query Console <http://localhost:8000/qconsole/>
 - System Summary <http://localhost:8001/>
 - MarkLogic REST Server <http://localhost:8100/>

GETTING STARTED

JSON

- <https://github.com/marklogic/marklogic-samplestack>

XML

- <https://github.com/rjrudin/ml-gradle>

```
gradle importSampleData
```

QUERY CONSOLE

The screenshot displays the MarkLogic Query Console interface. At the top, the browser address bar shows `localhost:8000/qconsole/`. The navigation bar includes links for **MarkLogic**, **Query Console**, **Configuration Manager**, **Monitoring**, and **Admin**. The user is logged in as **admin**.

The main workspace is divided into two panes. The left pane, titled **Query 2**, contains an XQuery script:

```
1 xquery version "1.0-ml";
2
3 import module namespace search =
4   "http://marklogic.com/appservices/search"
5   at " /MarkLogic/appservices/search/search.xqy";
6
7 search:search("Zaha")
```

The right pane, titled **Workspace**, shows a list of queries: **Query 1** and **Query 2**.


Below the query editor, there is a **Run** button and a set of tabs: **Result** (selected), **Auto**, **Raw**, **Profile**, and **Explorer**. The **Result** tab displays the query output:


sample-project-content (sample-project-modules: /, server: sample-project) 160000 Documents


Displaying 1 - 50 of 160000 | Page 1 of 3200


Document	Format	Properties	Collections
/import/eids-from-920001-to-930000/2-s2.0-65249097826/citedby.xml	E cited-by	(properties)	scopus
/import/eids-from-920001-to-930000/2-s2.0-66149132674/citedby.xml	E cited-by	(properties)	scopus
/import/eids-from-920001-to-930000/2-s2.0-67649376208/citedby.xml	E cited-by	(properties)	scopus
/import/eids-from-920001-to-930000/2-s2.0-70349983256/citedby.xml	E cited-by	(properties)	scopus
/import/eids-from-920001-to-930000/2-s2.0-70350741590/citedby.xml	E cited-by	(properties)	scopus
/import/eids-from-920001-to-930000/2-s2.0-76149110485/2-s2.0-76149110485.xml	E xocs:doc	(properties)	scopus
/import/eids-from-920001-to-930000/2-s2.0-76549136842/2-s2.0-76549136842.xml	E xocs:doc	(properties)	scopus


SYSTEM SUMMARY


 **Configure**


 Groups


 **Databases**


 App-Services


 Documents


 Extensions


 Fab


 Last-Login


 Meters


 Modules


 **sample-project-content**


 **Forests**

 Sub-Databases

 Flexible Replication

 Database Replication

 Fragment Roots

 Fragment Parents

Configure

Help

Configure Forests in a Database

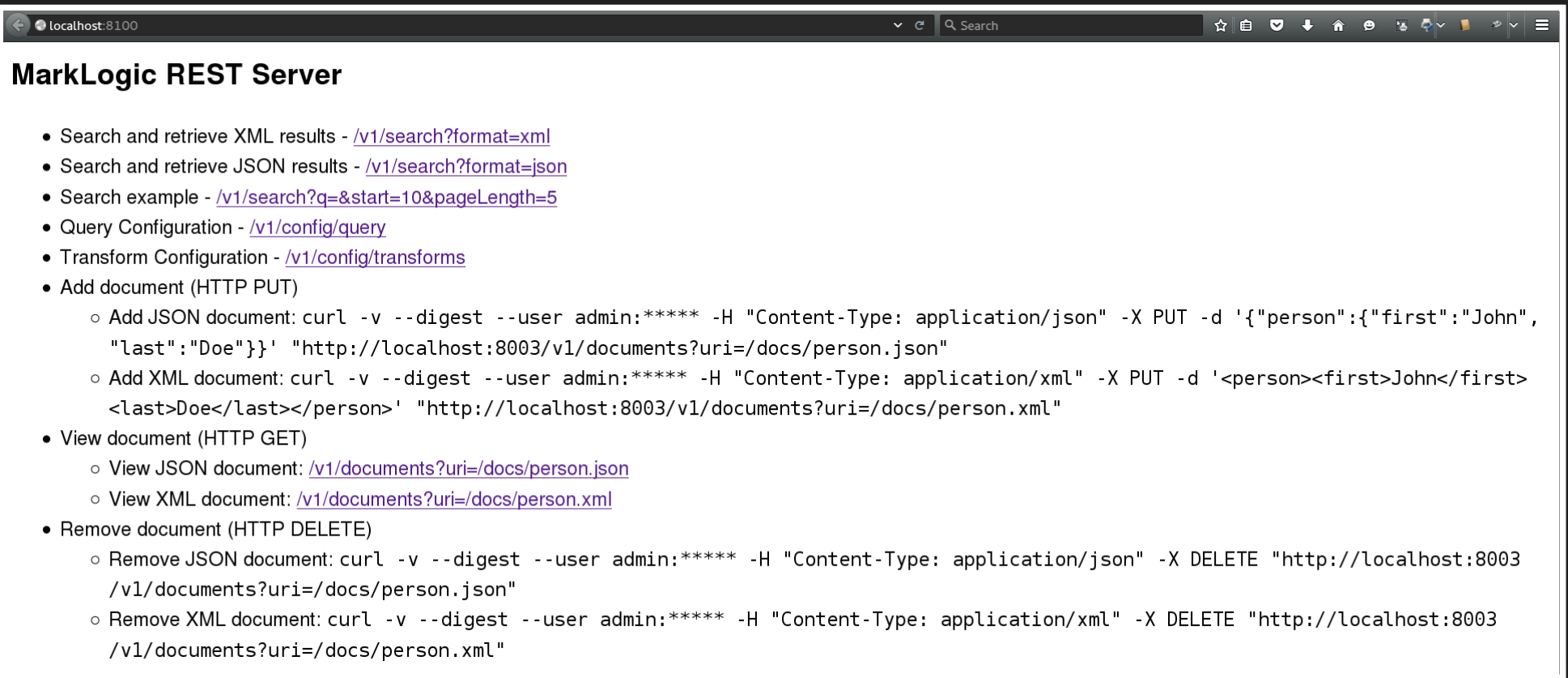
attached	retired	forest name
<input checked="" type="checkbox"/>	<input type="checkbox"/>	sample-project-content-1
<input checked="" type="checkbox"/>	<input type="checkbox"/>	sample-project-content-2
<input checked="" type="checkbox"/>	<input type="checkbox"/>	sample-project-content-3
<input checked="" type="checkbox"/>	<input type="checkbox"/>	sample-project-content-4

Attach AllDetach All

ok

cancel

REST SERVER



The screenshot shows a web browser window with the address bar set to `localhost:8100`. The page title is "MarkLogic REST Server". The content is a list of REST API endpoints and their usage examples.

MarkLogic REST Server

- Search and retrieve XML results - </v1/search?format=xml>
- Search and retrieve JSON results - </v1/search?format=json>
- Search example - </v1/search?q=&start=10&pageLength=5>
- Query Configuration - </v1/config/query>
- Transform Configuration - </v1/config/transforms>
- Add document (HTTP PUT)
 - Add JSON document: `curl -v --digest --user admin:***** -H "Content-Type: application/json" -X PUT -d '{"person":{"first":"John","last":"Doe"}}' "http://localhost:8003/v1/documents?uri=/docs/person.json"`
 - Add XML document: `curl -v --digest --user admin:***** -H "Content-Type: application/xml" -X PUT -d '<person><first>John</first><last>Doe</last></person>' "http://localhost:8003/v1/documents?uri=/docs/person.xml"`
- View document (HTTP GET)
 - View JSON document: </v1/documents?uri=/docs/person.json>
 - View XML document: </v1/documents?uri=/docs/person.xml>
- Remove document (HTTP DELETE)
 - Remove JSON document: `curl -v --digest --user admin:***** -H "Content-Type: application/json" -X DELETE "http://localhost:8003/v1/documents?uri=/docs/person.json"`
 - Remove XML document: `curl -v --digest --user admin:***** -H "Content-Type: application/xml" -X DELETE "http://localhost:8003/v1/documents?uri=/docs/person.xml"`

http://localhost:8100/v1/search?q=OECD&start=1&pageLength=5

```
localhost:8100/v1/search?q=OECD&start=1&pageLength=5
- <search:response snippet-format="snippet" total="490" start="1" page-length="5">
- <search:result index="1" uri="/import/eids-from-920001-to-930000/2-s2.0-70449625259/2-s2.0-70449625259.xml" path="fn:doc("/import/eids-
from-920001-to-930000/2-s2.0-70449625259/2-s2.0-70449625259.xml)" score="271.360" confidence="0.7620426" fitness="1" href="/v1
/documents?uri=%2Fimport%2Feids-from-920001-to-930000%2F2-s2.0-70449625259%2F2-s2.0-70449625259.xml" mimetype="application/xml"
format="xml">
- <search:snippet>
- <search:match path="fn:doc("/import/eids-from-920001-to-930000/2-s2.0-70449625259/2-s2.0-70449625259.xml)/*:doc/*:item/item/bibrecord
/head/citation-info/author-keywords/author-keyword[5]">
  <search:highlight>OECD</search:highlight>
</search:match>
- <search:match path="fn:doc("/import/eids-from-920001-to-930000/2-s2.0-70449625259/2-s2.0-70449625259.xml)/*:doc/*:item/item/bibrecord
/head/abstracts/abstract/*:para">
  In this article we attempt to analyse how
  <search:highlight>OECD</search:highlight>
  knowledge production is integrated with the process in which Finnish education policy takes shape.... ..the uses of the
  <search:highlight>OECD</search:highlight>
  PISA Study...
</search:match>
- <search:match path="fn:doc("/import/eids-from-920001-to-930000/2-s2.0-70449625259/2-s2.0-70449625259.xml)/*:doc/*:item/item/bibrecord
/tail/bibliography/reference[3]/ref-info/ref-title/ref-titletext">
  ...the
  <search:highlight>OECD</search:highlight>
</search:match>
</search:snippet>
</search:result>
```

R SHINY USER INTERFACE

- create query URL based on user inputs
- create text string for each match in parsed JSON results

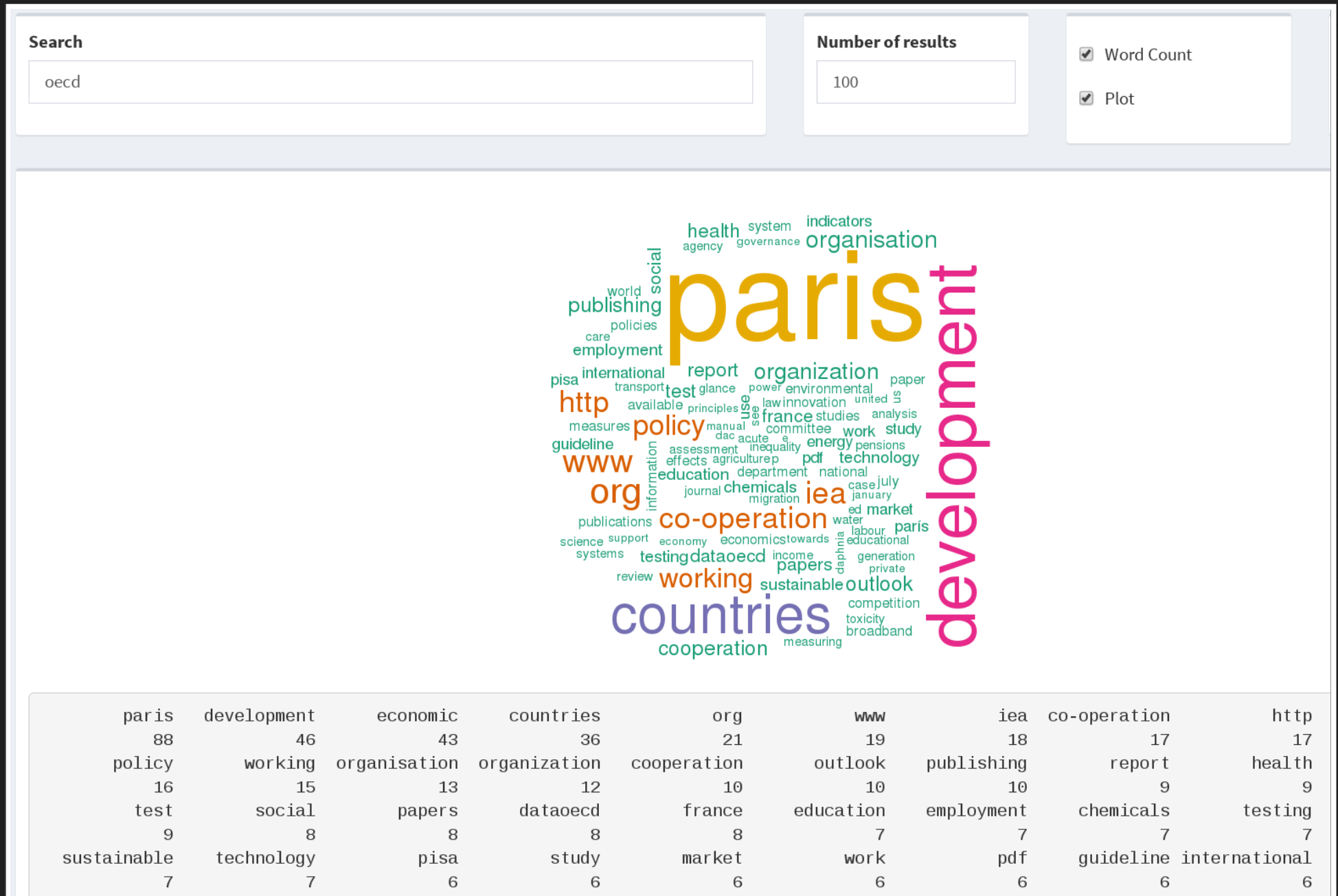
```
.search_result <- reactive({  
  url <- paste0('http://localhost:8100/v1/search?q=', input$search_  
               '&pageLength=', input$search_pagelength, '&format=')  
  tt <- getURL(url, userpwd="admin:admin")  
  result_df <- fromJSON(tt)$results  
  search_result <- sapply(result_df$matches, function(x) toString(u  
  names(search_result) <- unname(sapply(result_df$uri, basename))  
  return(search_result)  
})
```

QUANTEDA

R functions for Quantitative Analysis of Textual Data

- create corpus containing the document level information
 - dfm creates a document-feature matrix
- topfeatures returns a list of words sorted by frequency

R SHINY UI: WORD CLOUD, WORD COUNT



NOSQL OPEN-SOURCE

- eXistdb
- MongoDB
- Apache CouchDB