|  |
| --- |
| Dansk Data Arkiv |
| DDI indekseringsplatform og søgemaskine |
| Implementeret som eXist DB med Lucene som indeks |

|  |
| --- |
| Kemal Pajevic  08-01-2012 |

Indhold

2 Introduktion 1

2.1 Formål 1

2.2 eXist DB og Lucene 1

3 Opbygning af projektet 1

3.1 Filer og mapper 1

3.2 Kodebasen 2

4 Opsætning 3

5 Denormalisering 4

6 Index 4

6.1 Full-text indeks 4

6.2 Range indeks 5

7 Søgning 5

7.1 Format af søgestreng 5

7.2 Analyzer 6

7.3 Word stemmer 6

8 URN resolution 7

# Introduktion

## Formål

Formålet med projektet er at opsætte en XML database indeholdende studiedata i DDI format. Der skal efterfølgende opbygges indekser over udvalgte elementer i formatet for at effektivisere søgningen i dem, og endelig skal der implementeres søgefunktioner for at udtrække ønsket data fra databasen på baggrund af søgetekst samt nogle beskrivende parametre.

## eXist DB og Lucene

Til indekseringen af data har man valgt Lucene[[1]](#footnote-1) og eXist DB[[2]](#footnote-2) blev valgt som databasesystem. De er begge implementeret i Java og eXist bruger Lucene som sin default indekser. Interaktionen til eXist foregår igennem query sproget XQuery.

Den seneste udgivne version af eXist er 1.4, men denne indeholder en ældre version af Lucene som ikke inkluderer en såkaldt word analyzer til dansk. Derfor foregår udviklingen af dette projekt i en speciel branch af eXist projektet som findes på

<https://exist.svn.sourceforge.net/svnroot/exist/branches/adam/eXist-lucene-3.0>

En version 1.5 af eXist er i skrivende stund på trapperne, og denne bliver opdateret med den nyeste version af Lucene. Alternativt kan man skifte til hoved-branchen af eXist projektet når denne lucene-3.0 branch bliver merget ind.

# Opbygning af projektet

## Filer og mapper

Projektmappen er delt op i følgende undermapper:

|  |  |  |
| --- | --- | --- |
| **Mappenavn** | **Indeholder** | **Del af kodebasen** |
| dda | Filer til søgning i DDI databasen | Ja |
| dda-denormalization | Filer til denormalisering af databasen | Ja |
| dda-urn | Filer til URN resolution | Ja |
| doc | Dokumentation af projektet som Word dokument samt html-baseret beskrivelse af moduler og funktioner (á la Java docs) | Nej |
| JavaRestClientDDA | Java testklient til test af projektets REST interface | Nej |
| xml-sample-instances | Eksampel XML for typer anvendt i projektet | Nej |

Tabel 1 - Oversigt over mapper i projektet

## Kodebasen

Filer til søgning i DDI databasen (**dda**):

|  |  |  |  |
| --- | --- | --- | --- |
| **Filnavn** | **Beskrivelse** | | **Beliggenhed i eXist** |
| **index** | | | |
| collection.xconf | Indeks-konfigurationsfil til hovedcollection som indeholder DDI data | | /db/system/config/db/dda |
| **lib** | | | |
| context-functions.xquery | Funktioner til ekstrahering og trimming af det kontekst som matches blev fundet i | | /db/dda/lib |
| result-functions.xquery | Funktioner til behandling og indpakning af søgeresultater | | /db/dda/lib |
| search.xquery | Søgefunktioner | | /db/dda/lib |
| **rest** | | | |
| advanced-search.xquery | REST interface for avanceret søgning | | /db/dda/rest |
| simple-search.xquery | REST interface for simpel søgning | | /db/dda/rest |
| **schema** | | | |
| result/ ddieditor-lightxmlobject.xsd | Formatet som resultater returneres i | Lægges ikke ind | |
| result/ result-metadata.xsd | Beskriver resultatet (antal hits, antal sider, aktuel side, osv.) | Lægges ikke ind | |
| search/ advanced-search-parameters.xsd | Formatet af parameteret til avanceret søgning | Lægges ikke ind | |
| search/ scope.xsd | Angiver ønskede DDI elementer (enten dem som ønskes fremsøgt eller dem som ønskes vist som referencer) | Lægges ikke ind | |
| search/ search-metadata.xsd | Beskriver søgningen (ønsket antal resultater pr. side, anmodet side, osv.) | Lægges ikke ind | |
| search/ simple-search-parameters.xsd | Formatet af parameteret til simpel søgning | Lægges ikke ind | |

Tabel 2 - forklaring og placering af filer i *dda* mappen

Filer til denormalisering af databasen (**dda-denormalization**)

|  |  |  |
| --- | --- | --- |
| **Filnavn** | **Beskrivelse** | **Beliggenhed i eXist** |
| denormalize.xquery | Opbygger denormaliseringsdata som bruges til optimering af søgningen | /db/dda-denormalization |
| **index** | | |
| collection.xconf | Indeks-konfigurationsfil til den denormaliserede collection | /db/system/config/db/dda-denormalization |
| **schema** | | |
| denormalized-ddi.xsd | Formatet af den denormaliserede collection | Lægges ikke ind |

Tabel 3 - forklaring og placering af filer i *dda-denormalization* mappen

Filer til URN resolution (**dda-urn**)

|  |  |  |
| --- | --- | --- |
| **Filnavn** | **Beskrivelse** | **Beliggenhed i eXist** |
| **index** | | |
| collection.xconf | Indeks-konfigurationsfil til URN resolution collection | /db/system/config/db/dda-urn |
| **lib** | | |
| urn.xquery | Funktioner til URN resolution | /db/dda-urn/lib |
| **rest** | | |
| urn-resolution.xquery | REST interface for URN resolution | /db/dda-urn/rest |

Tabel 4 - forklaring og placering af filer i *dda-urn* mappen

Mere detaljeret beskrivelse af moduler og funktioner kan findes i **doc** mappen.

# Opsætning

Der skal oprettes tre collections i roden (dvs. i den overordnede **/db** mappe) af eXist DB:

* **dda** - Her uploades DDI XML’erne. Deres placering i collectioen er underordnet, men det anbefales at de samles i en bestemt undercollection, f.eks. **data** eller **ddi** (navnet på dette undercollection er ligegyldigt ift. søgningen).
* **dda-denormalization** - Her bliver det denormaliserede data gemt når denormaliseringen eksekveres.
* **dda-urn** - Dette er en særskilt collection som indeholder DDI data til URN resolution som kan indeholde flere udgaver af DDI elementerne.

I indeks-konfigurationsmappen (**/db/system/config/db**) oprettes der tre tilsvarende collections hvor de tilhørende konfigurationsfiler (**collection.xconf**) lægges.

Filerne fra projektets kodebase uploades også til disse collections i lokationerne specificeret i Tabel 2, Tabel 3 og Tabel 4. Undermapperne i stierne oprettes som undercollections.

Husk at hvis indeks-konfigurationsfilerne lægges ind (eller ændres) mens der findes data i den tilhørende collection, så skal denne collection reindekseres. Data som er lagt efter at indekset er på plads indekseres automatisk.

Det er vigtigt at køre denormaliseringen efter data er lagt ind, for ellers findes ingen referencer ved søgningen.

Kontekstmodulet (**context-functions.xquery**) anvender et bibliotek som hedder FunctX[[3]](#footnote-3). Det er ikke med i eXist som default, men da det er et meget populært XQuery bibliotek er der lavet et nemt installationsscript. Derfor er det nødvendigt at eksekvere scriptet som ligger i

<http://localhost:8080/exist/repo/install-functx.xqy>

eXist skal genstartes en gang før modulet bliver loadet.

# Denormalisering

For at nedsætte søgetiden er der implementeret en denormaliseret database som for samtlige relevante DDI elementer (Concept, Universe, QuestionItem, MultipleQuestionItem, Variable, Category) indeholder samtlige relevante referencer, dvs. elementer som enten refereres af eller referer til dette element.

For at opbygge denormaliseringsdata eksekveres scriptet **/db/dda-denormalization/denormalize.xquery**.

Denne proces sletter alt eksisterende data og genopbygger fra bunden.

Det er vigtigt at udføre processen med faste intervaller (f.eks. som et skeduleret job som køres en gang om ugen) for at holde data opdateret. Den forventede (beregnet på baggrund af udførsel på mindre datasæt) køretid for processen med ca. 2500 studier i databasen er på ca. 2 timer.

# Index

Indeks-konfigurationsfilen (**collection.xconf**) er stedet hvor man fortæller eXist hvilke elementer man ønsker at indeksere. Der laves én konfigurationsfil pr. collection og placeres i eXist som specificeret i Tabel 2, Tabel 3 og Tabel 4. Indekseringen kan generelt foregå på to måder, med et ”full-text” indeks eller et ”range” indeks.

## Full-text indeks

eXist anvender Lucene til at indeksere fritekst. Denne konfigureres i **lucene** elementet i konfigurationsfilen. Der specificerer man hvilke word analyzers man vil bruge samt de enkelte elementer man vil indeksere:

<analyzer id="da" class="org.apache.lucene.analysis.da.DanishAnalyzer"/>

**class** henviser til en Java klasse i Lucene biblioteket.

I dette projektet anvendes WhitespaceAnalyzer som er en helt basis analyzer der ikke foretager nogen behandling af teksten, men søger på den direkte tekststreng. Dette gør den velegnet til f.eks. at søge i id’er. Derudover anvenders DanishAnalyzer som er en analyzer specifikt til det danske sprog. Den filtrerer almindelige danske ord (som er irrelevante for søgningen) fra, og har desuden en såkaldt ”stemmer” som finder stammen af ord, og derfor kan søge på ord uafhængigt af bøjninger. Mere om dette findes i afsnit 7.2 samt 7.3.

De indekserede elementer specificeres med et **text** element:

<text qname="r:Title" analyzer="da" boost="6.0"/>  
<text match="//su:Abstract/r:Content" analyzer="da" boost="3.0"/>

Attributten **qname** bruges når vi indekserer på et qualified name, and **match** bruges når vi gerne vil specificere en sti til elementet for at adskille den fra et element med samme qualified name, men som ligger et irrelevant sted. Vi angiver desuden hvilken analyzer der bruges. Til sidst kan man specificere et boost værdi. Det er nemlig sådan at eXist angiver en score for alle søgeresultater på baggrund af hvor relevant den mener at resultatet er ift. søgeteksten, og denne score bruges så til at sortere resultaterne før vi præsenterer dem for brugeren. Vi kan manuelt booste en score ved at gange den med værdien angivet i **boost** attributten. På den måde fortæller vi eXist at vi f.eks. prioriterer resultater fundet i titlen højere end dem fundet i abstract. Denne attribut er valgfri og default værdien er 1.0.

For at bruge Lucene indexet ved en query anvendes funktionen

*ft:query($nodes as node()\*, $query as item()) node()\*.*

f.eks.:

//su:StudyUnit[ft:query(r:Citation/r:Title, “søgetekst”)]

hvor første argument er elementet man søger i, angivet relativt til det element man ønsker at få returneret (her StudyUnit).

Denne type indeks er mest velegnet til tekst.

## Range indeks

Range indeks er en simplere indeks som kun kan angives på en qualified name og kræver at man angiver typen på elementet:

<create qname="r:StartDate" type="xs:dateTime"/>

Fordelen ved denne indeks er at den kan bruges til både at matche præcise værdier, men også til at finde elementer hvis værdi er større eller mindre end en bestemt værdi.

Dette gør den velegnet til tidspunkter, datoer, tal, osv.

Range indeks anvendes med helt almindelig XPath syntakst:

//su:StudyUnit[r:Coverage/r:TemporalCoverage/r:ReferenceDate/r:StartDate >= $someDate]

# Søgning

Søgefunktionaliteten udføres af to funktioner, ddi:simpleSearch og ddi:advancedSearch. Funktionerne tager en XML element som parameter specificeret i hhv. **simple-search-parameters.xsd** og **advanced-search-parameters.xsd**. Disse XML elementer er beskrevet i detaljer i HTML dokumentationen for **search.xquery** modulet, og forklares derfor ikke her. For begge funktioner returneres resultaterne pakket ind i XML i formatet specificeret i **ddieditor-lightxmlobject.xsd**.

## Format af søgestreng

Det er som del af søgestrengen muligt at bruge specielle modifiers som bruges til at tilpasse søgningen. Disse er givet i tabellen nedeunder. Hvor modifieren er et ord skal der bruges store bogstaver.

De forskellige modifiers kan alle kombineres med undtagelse af nogle få som er modstridende (f.eks. opprioritering og fravalg).

|  |  |  |
| --- | --- | --- |
| **Modifier** | **Forklaring** | **Eksempel** |
| AND | Hvis AND bruges mellem to ord kræves der at begge ord findes i teksten. | politisk AND parti |
| OR | Hvis ORD bruges mellem to ord kræves der at mindst et af ordene findes i teksten. Dette er default opførsel (hvis ingenting angives) | politisk OR parti |
| ” ” | Søgning på eksakte fraser. | ”politisk parti” |
| + og - | Opprioritering og fravalg af ord. ’+’ kræver at ordet **skal** indgå, mens ’-’ siger at ordet ikke må indgå i teksten. | +politisk -parti |
| \* | Søgning med wildcard. Erstatter et hvilken som helst understreng. | pol\*  po\*tik  \*litik |
| ( ) | Gruppering af ord og modifiers. | (politik OR parti) AND valg |

Tabel 5 – Søgemodifiers

## Analyzer

Som nævnt i afsnit 6 anvendes en DanishAnalyzer til at analysere teksten som indekseres og søges. Denne filtrerer nogle af de mest almindelige danske ord som ikke indeholder nyttig information relevant for en søgning, de såkaldte ”stop words”. Disse ord er:

og, i, jeg, det, at, en, den, til, er, som, på, de, med, han, af, for, ikke, der, var, mig, sig, men, et, har, om, vi, min, havde, ham, hun, nu, over, da, fra, du, ud, sin, dem, os, op, man, hans, hvor, eller, hvad, skal, selv, her, alle, vil, blev, kunne, ind, når, være, dog, noget, ville, jo, deres, efter, ned, skulle, denne, end, dette, mit, også, under, have, dig, anden, hende, mine, alt, meget, sit, sine, vor, mod, disse, hvis, din, nogle, hos, blive, mange, ad, bliver, hendes, været, thi, jer, sådan

Listen er ordnet, fra mest almindelige til mest sjældne.

Skulle man have behov for det, kan der laves en egen liste af stop words. Dette er forklaret i eXist dokumentationens kapitel ***Lucene-based Full Text Index***, afsnit ***3.4. Analyzers***[[4]](#footnote-4).

## Word stemmer

Udover en dansk analyzer er Lucene også udstyret med en såkaldt ”word stemmer” til dansk. Denne finder stammen af ord og kan derfor søge på tværs af ordbøjninger.

Den kan f.eks. håndtere navneord således at det ikke betyder noget om der i teksten står ”bold”, ”bolde”, ”bolden”, ”boldens”, ”boldene”, ”bolderne”, osv. Ligeledes klarer den udsagnsord som ”lege”, ”leger”, ”legede”, ”leget”, ”leg”, osv.

Dog kan den tilsyneladende have problemer med visse irregulære bøjninger, så den f.eks. ikke genkender at ”cykel” og ”cykler” har samme ordstamme.

Stemmeren er aktiv som default.

# URN resolution

Udover den collection som anvendes til almindelig søgning er der implementeret en yderligere collection indeholdende DDI data. Denne hedder **dda-urn** og anvendes til URN resolution.

Til interaktion med denne collection er der implementeret en funktion urn:resolveUrn som tager en URN adresse[[5]](#footnote-5) som parameter.

Funktionen finder et unikt DDI element ved at først slå op på elementets ID. Da der kan findes flere elementer med det samme ID (hvis elementet har været ændret og derfor findes i flere versioner) sammenligner den desuden elementets versionsnummer med det som er specificeret i adressen og returnerer det rigtige element. Hvis elementet ikke selv har en version attribut kigges der efter versionen i elementets parent element, eller parents parent, osv. indtil atributten er fundet.

1. [http://lucene.apache.org](http://lucene.apache.org/) [↑](#footnote-ref-1)
2. <http://exist-db.org/> [↑](#footnote-ref-2)
3. <http://www.xqueryfunctions.com> [↑](#footnote-ref-3)
4. <http://exist-db.org/lucene.html#d1950e712> [↑](#footnote-ref-4)
5. Formatet af adressen er forklaret i HTML dokumentationen for modulet. [↑](#footnote-ref-5)