

Ekilex ja Sõnaveeb rakenduste paigaldusjuhend

TripleDev, Martin Laubre

Sisukord

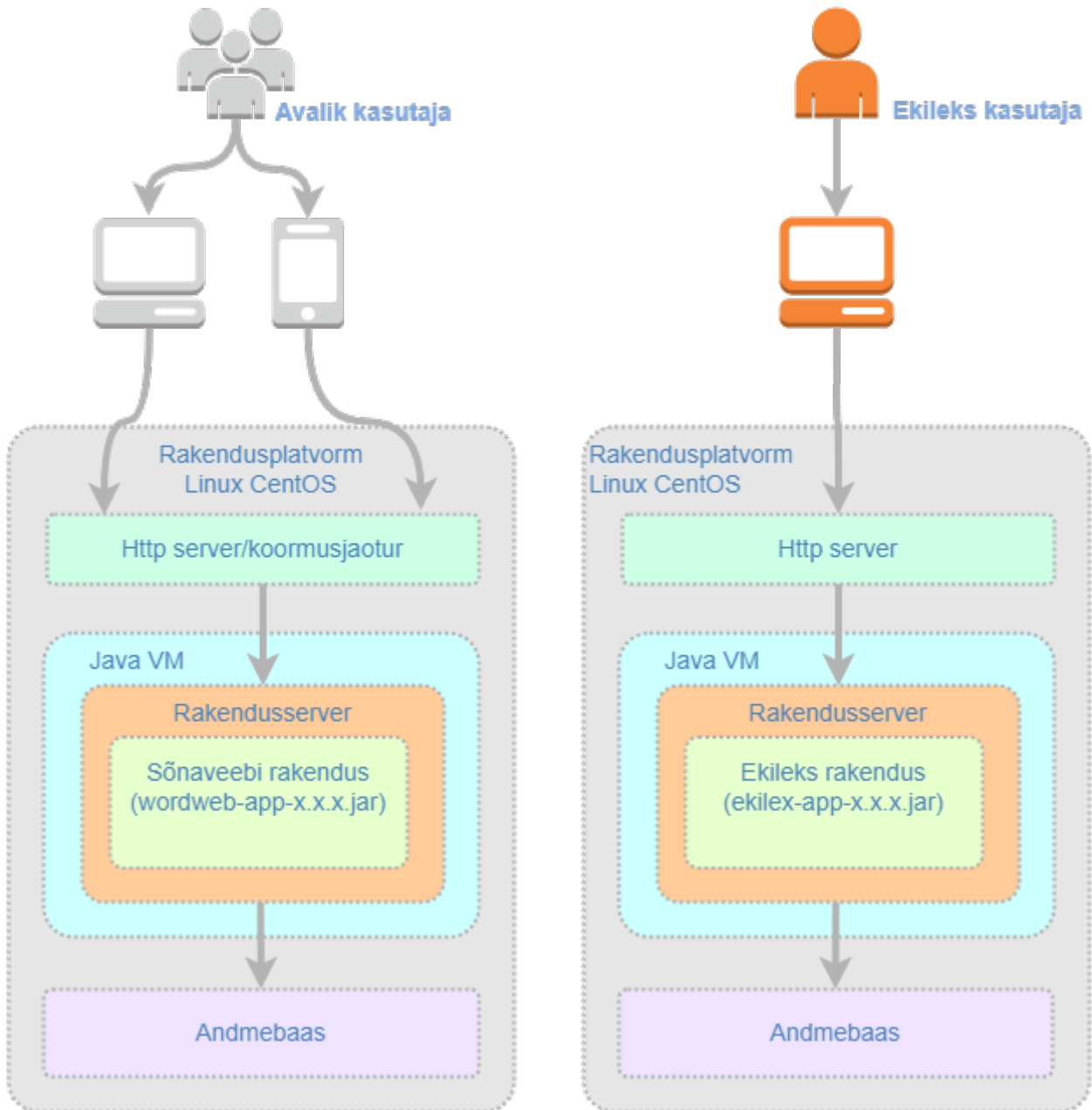
| | |
|---|----|
| Sissejuhatus..... | 4 |
| Tarkvara paigaldamine..... | 5 |
| Baas-tarkvara..... | 5 |
| Java..... | 5 |
| Apache HTTP | 5 |
| Apache Maven..... | 5 |
| Postgres..... | 6 |
| Apache Tomcat..... | 6 |
| Git..... | 7 |
| Rakendustarkvara..... | 7 |
| Tarkvaraprojekt Ekilex..... | 7 |
| Veebirakendus Ekilex..... | 8 |
| Lähtekood..... | 8 |
| Konfigureerimine..... | 8 |
| Andmebaasi struktuuride loomine..... | 10 |
| Ehitamine..... | 10 |
| Käivitamine..... | 10 |
| Kasutajate juurdepääs..... | 12 |
| Kasutaja kasutustingimustega nõustumine..... | 12 |
| Veebirakendus Sõnaveeb (Wordweb)..... | 12 |
| Lähtekood..... | 12 |
| Konfigureerimine..... | 13 |
| Andmebaasi struktuuride loomine..... | 14 |
| Ehitamine..... | 14 |
| Käivitamine..... | 15 |
| Andmete uuendamine..... | 16 |
| Statistika kogumise ja jagamise rakendus (Ekistat)..... | 17 |
| Lähtekood..... | 17 |
| Konfigureerimine..... | 17 |
| Andmebaasi struktuuride loomine..... | 18 |
| Ehitamine..... | 18 |
| Käivitamine..... | 18 |
| Sõnakogude laadurid..... | 18 |
| Andmetöötlusutiliidid..... | 19 |
| Homonüüminumbrite järjestamine..... | 19 |
| Sõnakogude eksportimine..... | 19 |
| Sõnakogude importimine..... | 19 |
| Ekspluatatsiooni märkused..... | 20 |
| Kasutajatega seotud kirjade varundamine sõnakogude täis-laadimisel..... | 20 |
| Rakenduse URL kodeeringute probleemistik..... | 20 |
| Sõnakogude haldus ja õigused..... | 20 |
| Uuenevate tarkvaraversioonide migratsioon..... | 21 |
| Lisad..... | 22 |

| | |
|--|----|
| Andmeparandusskriptid..... | 22 |
| Lihtsate vene vastete järjestuse nihutamine detailsetest tahapoolle..... | 22 |
| Tarbetu vene rõhu märgenduse eemaldamine..... | 22 |
| Homonüüminumbrite järjestamine..... | 22 |
| Tühjade ilmikute kustutamine..... | 23 |

Sissejuhatus

Järgnev juhend kirjeldab kahe autonoomse rakenduse, Ekilex ja Sõnaveeb ning nendega seotud baas-tarkvara paigaldamist.

Soovitav arhitektuur on ilmetatud järgneva joonisega:



Joonis 1. Arhitektuur

Kuna Ekilexil ja Sõnaveebil on erinevad kasutajad ning sellest tingitud erinevad süsteemi jõudluse eeldused, siis peavad rakendused asuma füüsiliselt erinevates serverites. Kummagi rakenduse jaoks tuleb teha eraldi täielik paigaldus baas-tarkvarast alates.

Sõnaveebi reaalsetest päringumahtudest tingitult on võimalik lisada jõudlust läbi andmebaasi ja rakenduse füüsilise lahutamise ning andmebaasi ja/või rakenduse klaster-konfiguratsiooni, kuid seda siis kui selleks reaalne vajadus tekib. Esialgu käesolev juhend sellist konfiguratsiooni ei käsitle.

Samuti ei käsitle käesolev juhend vabavaralise baas-tarkvaraga seotud spetsiifilisi määranguid ja häälestusi. Vastavad juhendid individuaalselt parima konfiguratsiooni koostamiseks on veebis vabalt leitavad.

Tarkvara paigaldamine

Baas-tarkvara

Java

Paigalda Java rakenduste virtuaalmasin Java JDK ver 8.
Operatsioonisüsteemile sobiv versioon leia siit:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

Apache HTTP

Rakendusserver on avaliku veebiliikluse eest varjatud Apache HTTP serveriga, mis klaster-konfiguratsiooni puhul käitub kui tarkvaraline koormusjaotur.
Operatsioonisüsteemile sobiv versioon leia siit:

<https://httpd.apache.org/download.cgi>

Kindlasti paigalda ja konfigureeri HTTP serveris SSL sertifikaat, et rakenduse veebiliiklus käiks üle https protokollile. Täienda hiljem HTTP serveri määranguid reaalsete rakenduse viidetega kui need ükskord paigaldatud saavad. Sulge igasugune muu juurdepääs rakendusserveritele. Rakenduse viite näidis:

```
<Location /wordweb>  
    ProxyPass          ajp://localhost:5577/wordweb nocanon  
    ProxyPassReverse    ajp://localhost:5577/wordweb  
</Location>
```

Apache Maven

Paigalda tarkvara ehitamise ja sõltuvuste halduse raamistik Apache Maven.

<https://maven.apache.org/download.cgi>

<http://www-eu.apache.org/dist/maven/maven-3/3.5.3/binaries/>

Käitu vastavalt juhendile:

<https://maven.apache.org/install.html>

Oluline on, et Maveni määrangud viitaksid korrektsele Java JDK-le.

Postgres

Paigalda andmebaasi tarkvara Postgres ver 9.6
Operatsioonisüsteemile sobiv versioon leia siit:

<https://www.postgresql.org/download/>

Ehkki Postgres andmebaasi vaikemäärangud on piisava jõudluse tagamiseks enamasti sobivad, siis saab jõudlust oluliselt tõsta andmebaasi kasutamise iseloomu ja reaalsel raudvara arvestades. Hea abivahend määrangute häälestamiseks asub siin:

<https://pgtune.leopard.in.ua/>

Loodud andmebaasi serveritesse loo vastava rakenduse andmebaas ja süsteemne kasutaja.

Rakendus Ekilex:

| | |
|--------------------------------|--------------------|
| Andmebaas: | ekilex |
| Kasutaja: | ekilex |
| Skeem: | public (vaikimisi) |
| Kodeering (<i>encoding</i>): | UTF8 |
| Tähestik (<i>collation</i>): | et_EE.UTF-8 |

Rakendus Sõnaveeb:

| | |
|--------------------------------|--------------------|
| Andmebaas: | wordweb |
| Kasutaja: | wordweb |
| Skeem: | public (vaikimisi) |
| Kodeering (<i>encoding</i>): | UTF8 |
| Tähestik (<i>collation</i>): | et_EE.UTF-8 |

Apache Tomcat

Rakendustarkvara rakendusserveriks on Apache Tomcat (<http://tomcat.apache.org/>). Antud juhul Tomcat serverit siiski eraldi paigaldama ei pea, sest rakendused Ekilex ja Sõnaveeb kasutavad Spring Boot Tomcat pistakut, mille vahendusel käivitatakse rakendus nõ virtuaalses Tomcat ümbrises (*wrapper container*).

Git

Paigalda tarkvara versioneerimise ja säilitamise tehnoloogia Git. Operatsioonisüsteemile sobiv versioon leia siit:

<https://git-scm.com/downloads>

Git kasutamise eelduseks on kasutajakonto olemasolu GitHub-s:

<https://github.com/>

Rakendustarkvara

Ekilex ja Sõnaveeb rakenduste lähtekoodi struktuur ning tarkvara ehitamise ja käivitamise mehhanism on samasugused. Seepärast on Ekilex rakenduse juhend detailsem ning Sõnaveebi juhend väldib korduvusi.

Tarkvaraprojekt Ekilex

Ekilex tarkvaraprojekti lähtekood asub siin:

<https://github.com/tripledev/ekilex.git>

Kui Git klientrakendus on konfigureeritud kasutama SSH privaatvõtit, siis alternatiivselt on lähtekoodi asukohaks:

`git@github.com:tripledev/ekilex.git`

Lae lähtekood kohalikus failisüsteemis kindlasse asukohta kuhu on mugav navigeerida ja tarkvara ehitamine käivitada. Ära kasuta tühikuid sisaldavate nimedega katalooge! Selle juhendi ulatuses on tinglikult selliseks kataloogiks

`/apps/source>`

Järgnev korraldus laeb alla kogu Ekilex tarkvara projekti, mille alamosadeks on Ekilex ja Sõnaveebi moodulid.

`/source>git clone https://github.com/tripledev/ekilex.git`

Navigeeri allalaetud tarkvara projekti juur-kataloogi `ekilex`

Allalaetud lähtekood esitab vaikumisi „master“ arendusharu seisu, mis pole ettenähtud toodangu režiimisi rakendamiseks. Toodangu jaoks on kokkulepitud stabiliseeritud versioon, mis asub mingis kindlas harus (*branch*) või *tag*-s. Sobivasse harusse ümberlülitumine käib käsuga

`/ekilex>git checkout x.x.x`

kus `x.x.x` on vastava haru nimi, mis tõenäoliselt on tarkvara versiooni number, näiteks `1.0.0`
Kõikide eksisteerivate harude nimekirja saab käsuga:

```
/ekilex>git branch
```

Ekilex ja Sõnaveeb rakenduste käivitamise eelduseks on keskkonna-spetsiifiliste määrangutega lähtekoodist ehitatud artefaktid

Veebirakendus Ekilex

Lähtekood

Ekilex rakenduse lähtekood asub tarkvara projekti alamkataloogis

```
/ekilex/ekilex-app>
```

Konfigureerimine

Peale lähtekoodi allalaadimist ja õigesse harusse ümberlülitumist, konfigureeri toodangukeskkonna-spetsiifiliste määrangute fail

```
/ekilex/ekilex-app/src/main/resources/application-prod.properties
```

Rakenduvad kõik määrangud `application.properties` ja valitud profiili (prod) laiendiga `application-prod.properties` faili peale kokku kusjuures profiili laiendiga fail on kõrgema prioriteediga. Seepärast ei ole vaja `application-prod.properties` failis kirjeldada määranguid, mis `application.properties` failis juba sobival kujul eksisteerivad. Samas saab `application-prod.properties` määrangutega kirjutada üle sobimatud määrangud `application.properties` failis. Olulised on järgmised parameetrid.

Rakendusserver:

```
server.port  
server.servlet.context-path  
server.servlet.session.timeout
```

```
tomcat.ajp.port  
tomcat.ajp.enabled
```

AJP määrangud reguleerivad rakendusserveri ja avalikku veebi eksponeeriva HTTP serveri vahelist kommunikatsiooni. AJP port peab kindlasti erinevama HTTP pordist. AJP protokoll kasutamine rakendusserveri ja HTTP serveri vahel on tungivalt soovitatav. Rakendusserveri määrangud peavad olema sellised, et need oleks võimalik avaliku veebi eest varjata.

Andmebaasi ühenduse kirjeldus:

```
spring.datasource.url
```



```
spring.datasource.username  
spring.datasource.password
```

Vastav loogiline andmebaas tuleb eelnevalt füüsilisse andmebaasi luua. Võimalik, et loogilise andmebaasi nimi ja kasutaja juba sobivadki. Samuti, kui andmebaas ja rakendus asuvad samas füüsilises serveris, võiks ka localhost url sobida.

Failiressursid – pildid, hääldused, jms. Sisu kohta küsi täpsemalt arendajatelt.

```
file.repository.path
```

Kasutajate registreerimisega seotud määrangud:

```
ekilex.app.url  
email.sending.enabled  
email.from.address  
email.from.name  
spring.mail.host  
spring.mail.port  
spring.mail.username  
spring.mail.password  
spring.mail.properties.mail.smtp.auth  
spring.mail.properties.mail.smtp.starttls.enable
```

Logimise detailsus:

```
logging.level.root  
logging.level.eki.ekilex  
logging.level.eki.common  
logging.level.org.jooq
```

Vaikimisi on rakenduse enda logid debug detailsusega, muud sõltuvused pääsevad läbi ainult warn taseme detailsusega. Alustuseks võib just selline logi detailsus olla isegi hea mõte.

Logifail:

```
logging.path  
logging.file
```

Failinime (logging.file) võiks muuta ainult juhul kui sellest rakendusest on plaan käivitada mitu erinevat instantsi. Reaalselt rikastatakse failinime veel kuupäevaga. Oluline on määrata logifailile asukoha absoluutne rada (logging.path) kuhu eri päevade logifailid kogunema hakkavad. Kui rada ei eksisteeri, see luuakse kui rakendust käivitaval kasutajal on selleks operatsioonisüsteemi mõistes vajalikud õigused.

Keskkondade eristamine:

`info.env.name`

Kui on soovi visuaalselt eristada Ekilexi versioone eri keskkondades, siis võib väärtustada lühidalt vastava keskkonna nimega (TEST, DEV, PREPROD1 jne). Pandud väärtus kuvatakse Ekilexi päises kollase (hoiatava) sildina. Vaikimisi on määranguks tühi väärtus, mis vastab toodangu keskkonnale kus sellist silti vaja pole.

Andmebaasi struktuuride loomine

Kõik andmebaasi struktuuride loomise ja baasandmetega täitmise skriptid asuvad kataloogis

```
/ekilex/eki-common/src/main/resources/sql/
```

Esmakordsel konfigureerimisel tuleb Sõnaveebi andmebaasis aktiveerida andmebaasi laiendid. See käib andmebaasi terminalis käsuga:

```
CREATE EXTENSION unaccent;
```

Ehitamine

Peale rakenduse konfigureerimist tuleb rakenduse lähtekoodist ehitada rakenduslik artefakt. Alustuseks tuleb ehitada kogu tarkvara projekt, mis loob ja laeb alla kohalikku Maven repositooriumisse nii selle tarkvaraprojekti moodulite artefaktid kui kõik nendele moodulitele vajalikud sõltuvused kesketest Maven repositooriumitest. Esmakordne ehitamine võib kesta minuteid.

```
/ekilex>mvn clean install -D skipTests
```

Ehitamine õnnestus kui protsessi lõpuks on kuvatud teade: „BUILD SUCCESS“

Ehitamise ebaõnnestumisest annab märku teade: „BUILD FAILURE“

Ebaõnnestunud ehitamise põhjus on kirjeldatud samas protsessi logis.

Edaspidi, kui on vaja muuta ainult vastava rakenduse määranguid, võib ehitamist käivitada selle rakenduse juures

```
/ekilex/ekilex-app>mvn clean install -D skipTests
```

Võti `skipTests` on vajalik kui kohalikult pole seadistatud automaatsete andmebaas. Igalpool mujal kui toodanguserveris, kus toimub lähtekoodi muutmine võiks automaatsete andmebaasi siiski konfigureerida ning ehitamine käivitada ilma `-D skipTests` määranguta, sest siis on eduka ehitamise protsessi eelduseks ka positiivse tulemusega automaattestid.

Käivitamine

Edukalt ehitatud rakenduse käivitamisel tuleb valida korrektne profiil (prod). Et kontrollida rakenduse konfigureerimise ja ehitamise edukust, saab seda testimiseks käivitada kahel viisil:

Maven pistaku vahendusel:

```
/ekilex/ekilex-app>mvn spring-boot:run -D spring-boot.run.profiles=prod
```

Otse Java rakendusena ehituse tulemi pealt:

```
/ekilex/ekilex-app/target>java -jar ekilex-app.jar --spring.profiles.active=prod
```

Kui rakendus käivitub normaalselt, on võimalik hakata jälgima rakenduse logi määratletud logifailis.

Selline rakenduse käivitusviis sobib siiski ainult ehitamise ja konfigureerimise edukuse testimiseks. Reaalselt on tungivalt soovitatav käivitada rakendus systemd teenusena.

Loo sobiv Linux kasutaja, kellena rakendust hakatakse käivitama. See ei tohiks olla root kasutaja!

Käesolevas juhendis on selleks kasutajaks **ekilex**

Kopeeri rakenduse ehituse tulemi kataloogist rakenduse artefakt **ekilex-app.jar** sobivasse kataloogi. Selle juhendi ulatuses on tinglikult selliseks kataloogiks

```
/apps/deploy/ekilex>
```

Loo fail

```
/etc/systemd/system/ekilex.service
```

Sisusta fail järgmisega:

```
[Unit]
Description=EKILEX application
After=syslog.target
```

```
[Service]
User=ekilex
ExecStart=/apps/deploy/ekilex/ekilex-app.jar
WorkingDirectory=/apps/deploy/ekilex
Environment="JAVA_HOME=/opt/jdk1.8.0_144"
SuccessExitStatus=143
```

```
[Install]
WantedBy=multi-user.target
```

Loo rakenduse artefakti naabrusesse täiendavate määrangute fail, mis kannab sama nime kuid on erineva faililaiendiga:

```
/apps/deploy/ekilex/ekilex-app.conf
```

Sisusta fail järgmisega:

```
JAVA_OPTS=-Xmx4096M  
RUN_ARGS=--spring.profiles.active=prod
```

Teenus on valmis kasutamiseks

```
/>systemctl start ekilex  
/>systemctl restart ekilex  
/>systemctl stop ekilex
```

Kasutajate juurdepääs

Ekilexi kasutamine on kaitstud autoriseerimisega. Kasutajad saavad ennast ise registreerida. Registreerimise protseduur sisaldab kinnituskirja saatmist kasutaja määratud e-posti aadressile. Kiri sisaldab linki, millega kasutaja oma registreerimist kinnitab. Kirja link on seotud ekilex rakenduse määrangutes kirjeldatud rakenduse aadressi.

Kasutaja kasutustingimustega nõustumine

Kasutaja registreerimisel küsitakse kasutajalt kinnitust Ekilexi kasutustingimustega. Kasutustingimustega nõustumine on registreerimise eelduseks. Kasutustingimuste tekst asub staatilisel kujul otse html-failis:

```
ekilex/ekilex-app/src/main/resources/view/html/terms.html
```

Kasutustingimuste versiooninumber on tekstilisel kujul rakenduse määrangute failis:

```
/ekilex/ekilex-app/src/main/resources/application-prod.properties
```

Võti:

```
terms.version=1.0
```

Vastav kehtiv väärtus salvestatakse kasutaja andmetesse kirje loomisel. Tulevikus saab luua selle baasil vajalik funktsionaalne loogika uuenevate kasutustingimuste versioonide kinnitamiseks endiste kasutajate jaoks.

Veebirakendus Sõnaveeb (Wordweb)

Lähtekood

Sõnaveebi rakenduse lähtekood asub tarkvara projekti alamkataloogis

```
/ekilex/wordweb-app>
```

Konfigureerimine

Konfigureeri toodangukeskkonna-spetsiifiliste määrangute fail

```
/ekilex/wordweb-app/src/main/resources/application-prod.properties
```

Rakendusserver:

```
server.port  
server.servlet.context-path  
server.servlet.session.timeout
```

```
tomcat.ajp.port  
tomcat.ajp.enabled
```

Rakendusserveri tööstatistika pistaku Actuator juurdepääsu kirjeldus:

```
spring.security.user.name  
spring.security.user.password  
spring.security.user.roles
```

Andmebaasi ühenduse kirjeldus:

```
spring.datasource.url  
spring.datasource.username  
spring.datasource.password
```

Failiressursid:

```
file.repository.path
```

Välisliidesed – kõnesüntesaator, kõnetuvastus, eesti ja vene korpused.

```
speech.synthesizer.service.url  
speech.recognition.service.url  
corpora.service.est.url  
corpora.service.est.corpname  
corpora.service.rus.url  
corpora.service.rus.corpname  
corpora.service.rus.username  
corpora.service.rus.api.key
```

Kasutajate tagasiside link ekilex suunas:

```
wordweb.feedback.service.url
```

Logimise detailsus:

```
logging.level.root
logging.level.eki.ekilex
logging.level.eki.common
logging.level.org.jooq
```

Logifail:

```
logging.path
logging.file
```

Andmebaasi struktuuride loomine

Sõnaveebi andmebaas koosneb peamiselt materialiseeritud vaadetest (*materialized views*), mis koostatakse üle andmebaasi side-ühenduse (*dblink*) Ekilexi andmete pealt.

Kõik andmebaasi struktuuride loomise ja andmetega täitmise skriptid asuvad kataloogis

```
/ekilex/wordweb-app/src/main/resources/sql/
```

`create_mviews.sql` – kõikide materialiseeritud vaadete loomine ja sisustamine sõnakogude andmetega ja klassifikaatoritega

`create_tables.sql` – kõikide tabelite loomine ja sisustamine mängude andmetega

Esmakordsel konfigureerimisel tuleb Sõnaveebi andmebaasis aktiveerida andmebaasi laiendid. See käib andmebaasi terminalis käsuga:

```
CREATE EXTENSION dblink;
CREATE EXTENSION fuzzystmatch;
CREATE EXTENSION pg_trgm;
```

Vajaliku side-ühendust kahe andmebaasi vahel saab testida käsuga:

```
SELECT dblink_connect('host=localhost user=ekilex password=<parool> dbname=ekilex');
```

Modifitseeri selle päringu parameetreid vastavalt reaalsele Ekilex konfiguratsioonile.

Edukas ühendus kahe andmebaasi vahel tagastab tulemuseks „OK“

Ehitamine

Kogu tarkvaraprojekti ehitamine:

```
/ekilex>mvn clean install -D skipTests
```

Sõnaveebi rakenduse ehitamine:

```
/ekilex/wordweb-app>mvn clean install -D skipTests
```

Käivitamine

Maven pistaku vahendusel:

```
/ekilex/wordweb-app>mvn spring-boot:run -D spring-boot.run.profiles=prod
```

Otse Java rakendusena ehituse tulemi pealt:

```
/ekilex/wordweb-app/target>java -jar wordweb-app.jar --spring.profiles.active=prod
```

Käivitamine teenusena. Vajalikud eeldused.

Kopeeri rakenduse artefakt `wordweb-app.jar` kataloogi:

```
/apps/deploy/wordweb>
```

Loo fail

```
/etc/systemd/system/wordweb.service
```

Sisusta fail järgmisega:

```
[Unit]
```

```
Description=WORDWEB application
```

```
After=syslog.target
```

```
[Service]
```

```
User=wordweb
```

```
ExecStart=/apps/deploy/wordweb/wordweb-app.jar
```

```
WorkingDirectory=/apps/deploy/wordweb
```

```
Environment="JAVA_HOME=/opt/jdk1.8.0_144"
```

```
SuccessExitStatus=143
```

```
[Install]
```

```
WantedBy=multi-user.target
```

Loo rakenduse artefakti naabrusesse fail:

```
/apps/deploy/wordweb/wordweb-app.conf
```

Sisusta fail järgmisega:

```
JAVA_OPTS=-Xmx4096M
```

```
RUN_ARGS=--spring.profiles.active=prod
```

Teenus on valmis kasutamiseks

```
/>systemctl start wordweb  
>systemctl restart wordweb  
>systemctl stop wordweb
```

Andmete uuendamine

Sõnakogude Sõnaveebi laadimine ning hiljem värskendamine toimub üle andmbaaside sideühenduse (DB link) andmebaasi vaadete (view) vahendusel. Andmebaaside sideühendamisest on juttu Sõnaveebi rakenduse konfigureerimise juures.

Sõnakogude algne laadimine toimub andmebaasi skriptiga, mis asub tarkvaraprojekti lähtekoodis:

```
/ekilex/wordweb-app/fileresources/sql/create_mviews.sql
```

See skript kustutab eksisteerivad, loob uued materialiseeritud vaated (*materialized views*) vastavate Ekilex andmebaasi vaadete pealt ning indekseerib need vaated. Modifitseeri skriptis andmebaasi sideühenduse parameetreid. Ära unusta lõpetada skript käsuga:

```
commit;
```

Edaspidi tuleb materialiseeritud vaadete loomise skripti `create_mviews.sql` käivitada alati kui Ekilexi või Sõnaveebi andmemudelisse tehakse muudatusi.

Kui aga Ekilexis muutuvad ainult andmed, siis piisab materialiseeritud vaadete värskendamisest:

```
refresh materialized mview_ww_classifier;  
refresh materialized mview_ww_collocation;  
refresh materialized mview_ww_dataset;  
refresh materialized mview_ww_definition_source_link;  
refresh materialized mview_ww_form;  
refresh materialized mview_ww_lexeme;  
refresh materialized mview_ww_lexeme_freeform_source_link;  
refresh materialized mview_ww_lexeme_relation;  
refresh materialized mview_ww_lexeme_source_link;  
refresh materialized mview_ww_meaning;  
refresh materialized mview_ww_meaning_freeform_source_link;  
refresh materialized mview_ww_meaning_relation;  
refresh materialized mview_ww_word;  
refresh materialized mview_ww_word_etym_source_link;  
refresh materialized mview_ww_word_etymology;  
refresh materialized mview_ww_word_relation;  
refresh materialized mview_ww_word_search;
```


Statistika kogumise ja jagamise rakendus (Ekistat)

Moodul ekistat-app tegeleb statistika andmete (nt Sõnaveebi otsingu andmed) kogumise ja jagamisega. Mooduli arendus on alles alguses ning funktsionaalsus mittetäielik. Teistel rakendustel puudub sõltuvus sellest moodulist, tarkvara paigaldamine ei ole kohustuslik.

Lähtekood

Rakenduse lähtekood asub tarkvara projekti alamkataloogis

```
/ekilex/ekistat-app>
```

Konfigureerimine

Konfigureeri toodangukeskkonna-spetsiifiliste määrangute fail

```
/ekilex/ekistat-app/src/main/resources/application-prod.properties
```

Rakendusserver:

```
server.port  
server.servlet.context-path
```

Liidestuvate rakenduste turvavõtmed:

```
ekistat.service.wordweb.api.key  
(ekistat.service.ekilex.api.key)
```

Andmebaasi ühenduse kirjeldus:

```
spring.datasource.url  
spring.datasource.username  
spring.datasource.password
```

Logimise detailsus:

```
logging.level.root  
logging.level.eki.stat  
logging.level.eki.common  
logging.level.org.jooq  
logging.file.name
```

Sõnaveebi määrangute failis konfigureerida

```
/ekilex/ekistat-app/src/main/resources/application-prod.properties
```

Statistika rakenduse andmed:

```
ekistat.service.enabled  
ekistat.service.url  
ekistat.service.key
```

Andmebaasi struktuuride loomine

Kõik andmebaasi struktuuride loomise ja andmetega täitmise skriptid asuvad kataloogis

```
/ekilex/stat-app/src/main/resources/sql/
```

`create_tables.sql` – kõikide tabelite loomine

Ehitamine

Kogu tarkvaraprojekti ehitamine:

```
/ekilex>mvn clean install -D skipTests
```

Statistika rakenduse ehitamine:

```
/ekilex/stat-app>mvn clean install -D skipTests
```

Käivitamine

Maven pistaku vahendusel:

```
/ekilex/stat-app>mvn spring-boot:run -D spring-boot.run.profiles=prod
```

Otse Java rakendusena ehituse tulemi pealt:

```
/ekilex/stat-app/target>java -jar stat-app.jar --spring.profiles.active=prod
```

Sõnakogude laadurid

(Kogu funktsionaalsus ekilex-etl moodulis on aegunud)

Andmetöötlusutiliidid

Homonüüminumbrite järjestamine

Loodud on andmebaasi protseduur, mis likvideerib samakujuliste keelendite korduvad homonüüminumbrid ning ühtlasi järjestab homonüüminumbrid ümber absoluutse eelistusega tõsta „EKI ühendsõnastik“-u (sss) kuuluvusega keelendid esimeseks ja korduva homonüüminumbri korral keelendite loomise järjekorra alusel.

Erineval viisil sõnakogude lisandumisel võib olla vajalik protseduuri korduvalt käivitada.

Lisad > Andmeparandusskriptid > Homonüüminumbrite järjestamine

Sõnakogude eksportimine

NB! Sõnakogude ekspordi utiliit on kolinud ekilex-etl moodulist ekilex-app moodulisse! Vastav utiliit ekilex-etl moodulis on aegunud!

Täpsem info:

<https://github.com/tripledev/ekilex/wiki/S%C3%B5nakogude-eksport-import>

Sõnakogude importimine

NB! Sõnakogude impordi utiliit on kolinud ekilex-etl moodulist ekilex-app moodulisse! Vastav utiliit ekilex-etl moodulis on aegunud!

Täpsem info:

<https://github.com/tripledev/ekilex/wiki/S%C3%B5nakogude-eksport-import>

Ekspluatatsiooni märkused

Kasutajatega seotud kirjade varundamine sõnakogude täis-laadimisel

Kui on soov täis-laadida kõik sõnakogud, aga säilitada kasutajatega ja muu haldusega seotud andmed, siis on soovitatav kopeerida järgmiste tabelite sisu (kui need selles versioonis eksisteerivad):

```
eki_user, eki_user_application, eki_user_profile, dataset_permission
```

Et antud kirjade taastamisel säiliks korrektne relatsioonilisus, siis tuleb teadlikult juhtida PK (*Primary Key*) genereerimise ja ise määramise vahel arvestusega, et rakendus ise loob kirjeid kasutades automaatsed PK genereerimise jadasid (*sequence*). Seepärast peab vastavaid jadasid üle-häälestama juhul kui PK-sid on sisestatud käsitsi.

Üks lahendus on varundada nimetatud tabelite kirjed *insert* klauslitena ilma PK väärtusteta sorteerituna kohaliku PK järgi.

Oluline! Kui olemasolevas andmebaasis puudub tabel `eki_user_profile`, siis tuleb see käsitsi luua ja sisustada peale kasutajate laadimist skriptiga:

```
insert into eki_user_profile (user_id) (select id from eki_user);
```

Rakenduse URL kodeeringute probleemistik

Selleks, et rakenduse URL rajad saaksid sisaldada nõ eksootilisi ja reserveeritud sümboleid, peab rakendus vastavaid väärtuseid teadlikult kodeerima enne URL-i lisamist ja dekodeerima pärast URL-st lugemist. Paraku erinevad teegid ja filtrid võivad kasutada erinevaid kodeerimise reegleid ning lisaks rakendada iseseisvalt kodeerimist tingimuslikult sõltuvalt millised sümboolid URL-s sisalduvad vaatamata sellele, et URL on juba kodeeritud. Oleme märganud, et Apache HTTP server võib teatud tingimustel meelevaidselt ja soovimatult manipuleerida URL-e (eriti vene keelsete väärtustega URL-de puhul), kui selle määrangutes vastavat keeldu pole.

Soovitame täiendada HTTP serveri rakenduste viidete määranguid „nocanon“ võtmega:

```
...  
ProxyPass ajp://localhost:5577/wordweb nocanon  
...
```

Sõnakogude haldus ja õigused

Kõikidel laetud sõnakogudel tuleb enne kasutuselevõttu ja muutmist/täiendamist Ekilexis „Sõnakogude haldus“ lehel lisada kasutatavad klassifikaatorid. Vastavate klassifikaatorite valikud vastavates sõnakogudes on sisu muutmisel/täiendamisel piiritletud just selliste väärtustega. Vastasel juhul on valikud tühjad.

Sõnakogude haldamiseks peab kasutajal olema kas administraatori staatus või vastava sõnakogu omaniku õigus. Sõnakogu omaniku õigus laiendab muutmise õigust.

Sõnakogu sisu muutmiseks peab olema valitud vastava sõnakogu muutmise roll. Muutmise rolli on võimalik valida ainult juhul kui kasutajale on antud „Õiguste haldus“ lehel vastav muutmise õigus. Iga kasutaja võib otsida ja näha kõigi sõnakogude sisu, mis on olekus „avalik“. Kasutaja saa näha ka muudes olekutes sisu, kui tal on vastava sõnakogu muutmise õigus.

Administraatori staatusega kasutaja saab näha kõigi sõnakogude kogu sisu sõltumata olekust, kuid ei saa muuta seda.

Uuenevate tarkvaraversioonide migratsioon

Kuna alates Ekilex tarkvaraversioonist 1.8.0 alates on tarkvara hakatud toodangus kasutama sihtotstarbepäraselt ning andmebaasi tekib uusi andmeid, mida sõnakogude täislaadimisega ei saa korvata, aga tarkvaraarendus jätkuvalt täiendab ja muudab andmebaasi struktuure, siis edaspidi on oluline jälgida kokkuleppelist uuenduste migratsiooni protsessi.

- Tarnemärkmed kirjeldavad olulisemaid muudatusi süsteemis ning seotud tegevusi nende juurutamiseks. Tarnemärkmed asuvad ekilex tarkvaraprojekti juurkataloogis failis:

`ekilex/delivery.txt`

- Enne uuendatud tarkvara käivitamist, jooksutada andmebaasis järgnev skript:

`ekilex/eki-common/src/main/resources/sql/delivery.sql`

Palun jälgige ka kommentaare selles skriptis. Võimalik, et muudatused tabelistruktuurides tingivad mingite muude skriptide taaskäivitamise.

Lisad

Andmeparandusskriptid

Lihtsate vene vastete järjestuse nihutamine detailsetest tahapoole

```
update lexeme l
  set order_by = nextval('lexeme_order_by_seq')
from (select l.id
      from lexeme l,
           word w
      where l.complexity = 'SIMPLE'
            and l.word_id = w.id
            and w.lang = 'rus'
      order by l.order_by) lqq
where l.id = lqq.id;
```

Tarbetu vene rõhu märgenduse eemaldamine

```
update form
  set value_prese = replace(value_prese, '<eki-stress>ë</eki-stress>', 'ë')
where value_prese like '%<eki-stress>ë</eki-stress>%';
```

Homonüüminumbrite järjestamine

```
-- ajutine andmetüüp
create type temp_word_data_tuple as (word_id bigint, homonym_nr integer);

do $$
<<adj_homon_nr_block>>
declare
  ordered_homonym_nrs_str_pattern text := array_to_string(array (select
generate_series(1, 100)), '- ', '');
  word_row record;
  adj_word_ids temp_word_data_tuple;
  homonym_nr_iter integer;
begin
  for word_row in
    (select w.value,
           w.lang,
           w.word_ids
    from (select w.value,
                w.lang,
                array_agg(row(w.id, w.homonym_nr)::temp_word_data_tuple
order by w.ds_order_by, w.homonym_nr, w.id) word_ids,
                array_to_string(array_agg(w.homonym_nr order by
w.ds_order_by, w.homonym_nr), '- ', '') homonym_nrs_str
```

```
from (select w.id,
            w.value,
            w.homonym_nr,
            w.lang,
            (select case
                when count(l.id) > 0 then 1
                else 2
            end
            from lexeme l
            where l.word_id = w.id
            and l.type = 'PRIMARY'
            and l.dataset_code = 'sss') ds_order_by
from word w
where exists (select l.id
              from lexeme l
              where l.word_id = w.id
              and l.type = 'PRIMARY')) w
group by w.value,
         w.lang) w
where ordered_homonym_nrs_str_pattern not like w.homonym_nrs_str ||
'%'
order by w.lang,
         w.value)
loop
  homonym_nr_iter := 1;
  foreach adj_word_ids in array word_row.word_ids
  loop
    if homonym_nr_iter != adj_word_ids.homonym_nr then
      update word set homonym_nr = homonym_nr_iter where id =
adj_word_ids.word_id;
    end if;
    homonym_nr_iter := homonym_nr_iter + 1;
  end loop;
end loop;
end adj_homon_nr_block $$;
```

```
drop type temp_word_data_tuple;
```

Tühjade ilmikute kustutamine

```
delete
from lexeme
where id in (select l.id
              from lexeme l,
              meaning m
              where l.is_public = true
              and l.dataset_code != 'ety'
              and l.meaning_id = m.id
```

```
and not exists (select lf.id from lexeme_freeform lf where
lf.lexeme_id = l.id)
and not exists (select cl.id from lexeme_deriv cl where
cl.lexeme_id = l.id)
and not exists (select cl.id from lexeme_pos cl where
cl.lexeme_id = l.id)
and not exists (select cl.id from lexeme_region cl where
cl.lexeme_id = l.id)
and not exists (select cl.id from lexeme_register cl where
cl.lexeme_id = l.id)
and not exists (select lsl.id
                    from lexeme_source_link lsl
                    where lsl.lexeme_id = l.id)
and not exists (select lr.id from lex_relation lr where
lr.lexeme1_id = l.id)
and not exists (select lr.id from lex_relation lr where
lr.lexeme2_id = l.id)
and not exists (select l2.id
                    from lexeme l2
                    where l2.meaning_id = m.id
                    and l2.id != l.id)
and not exists (select lt.id from lexeme_tag lt where
lt.lexeme_id = l.id)
and not exists (select lc.id from lex_colloc lc where
lc.lexeme_id = l.id)
and not exists (select mf.id
                    from meaning_freeform mf
                    where mf.meaning_id = m.id)
and not exists (select mst.id
                    from meaning_semantic_type mst
                    where mst.meaning_id = m.id)
and not exists (select mr.id
                    from meaning_relation mr
                    where mr.meaning1_id = m.id)
and not exists (select d.id from definition d where
d.meaning_id = m.id)
and not exists (select md.meaning_id
                    from meaning_domain md
                    where md.meaning_id = m.id))
```

Juhendi lõpp

(Dokument täieneb jooksvalt. Jälgi dokumendi kuupäeva!)