HAIBA

EPIMIBA-behandling

Design, arkitektur og udviklerguide

Indhold

1 Formål 3

2 Arkitektur- og designoverblik 4

3 Særlige forhold 6

3.1 Personhenførbare data 6

4 Fysiske Datamodeller 7

4.1.1 Database 7

4.1.2 Meta-database 7

5 Opsætning af udviklingsmiljø 8

5.1 Kildekode 8

5.2 Byggemiljø 8

5.2.1 Dependencies 8

5.3 Database-setup 8

5.3.1 Database 8

5.4 Test 8

5.5 IDE 9

5.5.1 Eclipse 9

5.6 IntelliJ Idea IDE 10

5.7 Distribution 11

6 Tips og tricks 12

6.1 Tomcat out of memory 12

6.1.1 Beskrivelse 12

6.1.2 Løsning 12

7 Referencer og kilder 13

8 Ændringslog 14

# Formål

Dette dokument giver et overblik over EPIMIBA-databehandleren med fokus på design og arkitektur. Dokumentet er også en guide til udviklere. Guiden gennemgår på overordnet plan de aktiviteter, der er nødvendige for at kunne videreudvikle på servicen.

Dokumentet har som formål at give et indblik i det underliggende design, de udstillede snitflader på det overordnede niveau samt særlige forhold.

Både udviklere, aftagere samt driftsleverandør kan med fordel læse dette dokument.

# Arkitektur- og designoverblik

EPI-MIBA er en proxy service for MiBa databasen, som indeholder mikrobiologiske data fra alle kliniske mikrobiologiske afdelinger i Danmark. Mikrobiologiske data er en forudsætning for at kunne udrede HAI typerne: bakteriæmier, luftvejsinfektioner, postoperative sårinfektioner samt urinvejsinfektioner.

EPI-MIBA-dataimporteren sørger for, at indlæse data fra EPI-MIBA, og lægge data i HAIBA databasen som HAIBA projektet så kan tilgå.

Samtidig indeholder importeren en notifikationsmekanisme, der kan sende en email hvis der mødes nye infektionstyper eller lokationer

Overordnet set består importeren af en applikation kørende på en JEE applikationsserver (I dette tilfælde Tomcat), som læser en række stamdata samt prøvesvar fra en webservice og eksporter prøvesvar samt nye klassifikationer til 2 databaser; hhv. HAIBA og klassifikationsdb (kan være den samme afhælngig af konfigurtion).

Databehandleren er bygget op via Spring version 3.x (se <http://www.springsource.org/> ) som en standard JEE webapplikation. Kendskab til følgende Spring undermoduler vil være en fordel (MVC, batch og Integration).

Til at bygge systemet bruges Maven 3.x.

Modulerne er strukturerede på følgende måde:

* Dataimporteren har til ansvar at hente prøvesvar ind fra webservice (i pakken dk.nsi.haiba.epimibaimporter.ws), via batches af konfigurerbar størrelse, og placere det i en intern model baseret på POJO's.

Dataimporteren ligger i pakken: dk.nsi.haiba.epimibaimporter.importer, klassen ImportExecutor er det skedulerede job som sørger for at importere data når noget nyt dukker op. Dette ved at gemme det transaktionsid, der sidst blev importeret.

* Data hentes særskilt for hver prøvesvartype. Disse svartyper defineres manuelt i tabellen CaseDef, indholdet af denne tabel itereres ved hver kørsel.
* Efter hver kørsel kontrolleres de over tid importerede prøvesvar for nye mikroorganismer eller lokationer. Er der nye værdier i Klass\_microorganism eller Klass\_location, notificeres de på forhånd konfigurerede emailadresser med identifikation af de berørte nye data, som siden kan beriges i de nævnte tabeller.

Notifikationer om og kopiering af nye data håndteres i dk.nsi.haiba.epimibaimporter.importer.ImportExecutor.checkAndSendEmailOnNewImports.

* Degradering: Hvis databehandlingen afbrydes utilsigtet eks. pga. kodefejl eller fysisk afbrydelse, skal behandlingen forsætte uden tab ved næste kørsel. Dette varetages ved at gemme transaktionsid (pr. svartype) for sidst gemte prøvesvar i tabellen EpimibaTransaction. Transaktionsid antages at være unikt pr. prøvesvar. Ved næste kørsel hentes det sidst gemte transaktionsid og webservice spørges efter prøvesvar nyere end sidst gemte. Prøvesvar kan I værste tilfælde godt gemmes dobbelt i Header-tabellen.

# Særlige forhold

## Personhenførbare data

Data fra EPIMIBA indeholder personnumre, og derfor personhenførbare. Dette personnummer bæres igennem regelbehandling og efterfølgende gemt i HAIBA indlæggelsesdatabasen.

Personnumre vil aldrig blive skrevet i logfiler, alt data der logges er transactionid prøvesvarene, som gør at personer med relevant viden og rettigheder kan slå op i EPIMIBA og se data og derved sammenholde det med fejlbeskeden.

# Fysiske Datamodeller

HAIBA-databaserne indeholder tabeller med data, hvor Header er den overordnede tabel med prøvesvar. CaseDef indeholder de prøvesvartyper, der skal hentes prøvesvar for.

Detaljer omkring feltværdier, datatyper og lignende kan ses i SQL skemaerne som ligger på følgende URL: <https://github.com/trifork/HAIBA-EPIMIBAimporter/tree/master/database>

### Database

|  |  |
| --- | --- |
| Tabel | Beskrivelse |
| Header | Indeholder data for et prøvesvar. |
| CaseDef | Indeholder typer af prøvesvar, der skal efterspørges i webservice |
| Isolate | Indeholder bakterienumre refereret til prøvesvar |
| Quantitative | Indeholder mængdeangivelser refereret til prøvesvar |
| Tabmicroorganism | Klassifikationstabel for mikroorganismer |
| TabLabSection | Klassifikationstabel for laboratorier |
| TabOrganization | Klassifikationstabel for institutioner |
| TabAnalysis | Klassifikationstabel for analyse |
| TabInvestigation | Klassifikationstabel for undersøgelse |
| TabLocation | Klassifikationstabel for prøvelokationer |
| Klass\_microorganism | Anvendte mikroorganismer |
| Klass\_Location | Anvendte lokationer |

### Meta-database

|  |  |
| --- | --- |
| Tabel | Beskrivelse |
| EpimibaTransaction | Indeholder transactionid for seneste gemte prøvesvar, pr. type |
| EpimibaImporterStatus | Indeholder information om kørsler; start- og sluttid samt udfald. Desuden en kort tekstbesked om evt. fejl. Denne information bruges af statussiden. |

# Opsætning af udviklingsmiljø

Opsætningen af udviklingsmiljøet for LPR-databehandleren forudsætter, at følgende elementer allerede er installeret på udviklerens maskine:

* Java Developer Kit 6.0\_x
* Maven 3.x [MAVEN]
* Git 1.7.x
* MySQL 5.5.x
* Tomcat 7 (Udviklet og testet på version 7.0.34)

Installationsvejledningen indeholder detaljer omkring opsætning af ovenstående komponenter.

## Kildekode

Kildekoden er placeret i et github-repositorie og kan checkes ud på følgende måde:

git clone git@github.com:trifork/HAIBA-EPIMIBAimporter.git

## Byggemiljø

Databehandleren anvender Maven som byggesystem. Strukturen følger de generelle anbefalinger for Maven projekter, og er struktureret

efter Maven layout-konventionen.

For at bygge databehandlersystemet, skal man gøre følgende:

mvn install

Projektet indeholder udover unittests også integrationstests, der kræver en kørende database, for at undlade udførelsen af integrationstests kan parameteren -DskipITs tilføjes mvn install kommandoen.

### Dependencies

For at kunne hente NSI-specifikke afhængigheder (bl.a. nsp-util) i binær form i stedet for at skulle bygge alle afhængigheder selv på det lokale udviklingsmiljø, indeholder pom'en en reference til nexus.trifork.com, som er et artefaktrepository der er placeret hos Trifork. Binære releases af databehandleren findes også i nexus.trifork.com.

Repository’et bør, når det er muligt, udskiftes med et artefaktrepository der er driftet hos NSI. Når et sådant er etableret

## Database-setup

### Database

Databaseskema for prøvesvar mm. ligger i database-folderen.

## Test

Installationen kan verificeres ved at eksekvere databehandlerens test suite.

Testsuiten benytter JUnit og Mockito til test.

Test suiten afvikles ved at udføre følgende kommando i projektroden:

mvn test

Installationen kan yderligere verificeres (Code coverage, kode konventioner o.l.) ved at udføre kommandoen, rapporterne ligger i <projekt rod>/target/site:

mvn verify

## IDE

Databehandleren kan principielt udvikles i enhver Java IDE, der forstår Maven projekters opbygning.

I dette dokument beskrives kort opsætning for to af de pt. mest udbredte Java IDE’er: Eclipse og IntelliJ.

### Eclipse

Eclipse er ikke født med Maven support, og det anbefales derfor, at man installerer m2eclipse inden databehandleren hentes ind i Eclipse, pluginet kan hentes her:

http://www.eclipse.org/m2e/

Herefter importeres projekterne i Eclipse via ”import”:

Alternativt kan man importere projektet ved at udføre følgende kommando:

mvn eclipse:eclipse

Og herefter importere projektet på normal vis i Eclipse.

Kommandoen genererer Eclipse projektfilerne (.project og .classpath) for databehandleren. Denne metode kræver dog, at kommandoen udføres hver gang man ændrer i pom filerne.

## IntelliJ Idea IDE

IntelliJ Idea er født med Maven support, og databehandleren kan derfor direkte importeres. Projektet importeres i IntelliJ ved under ”Create new project” at vælge ”Import project from external model”. Herefter udvælges roden af databehandleren, hvorefter projektet importeres.

Det anbefales i den sammenhæng, at man krydser af i ”Import Maven projects automatically”, hvorefter IntelliJ selv detekterer nye moduler i projektet.

Alternativt kan man importere projektet ved at udføre følgende kommando:

mvn idea:idea

Herefter kan projektet importeres på normal vis i IntelliJ.

*Obs! Denne metode kræver dog, at kommandoen udføres hver gang man ændrer i pom filerne.*

## Distribution

databehandleren kan bygges til distribution eller lokal test ved at udføre:

mvn package

Dette generer en WAR fil, der efterfølgende kan deploye's lokalt eller på et testmiljø.

Til produktion bør Maven release plugin bruges, da det får tag'et bygget og automatisk får opdateret versionsnumre m.v.

Først køres:

mvn release:prepare

Går det godt køres:

mvn release:perform

# Tips og tricks

I det følgende beskrives problemer man som udvikler kan støde på, og forslag til løsning af samme.

## Tomcat out of memory

### Beskrivelse

I Tomcats log, logger den noget i stil med ”out of memory” og nævner “permgenspace”, dette kan ske ved at der hot-deployes ofte, som man typisk gør under udvikling

### Løsning

Forøg Tomcat permgen space ved at sætte følgende miljø variabel:

JAVA\_OPTS="-Xms2048m -Xmx2048m -XX:MaxPermSize=512m"

# Referencer og kilder

|  |  |  |
| --- | --- | --- |
| Reference-id | Indhold / Overskrift | Henvisning |
| [MAVEN] | Welcome to Apache Maven | <http://maven.apache.org/> |

# Ændringslog

Kilden til dette dokument kan findes på:

[https://github.com/trifork/HAIBA-EPIMIBAimporter/blob/master/doc/Design, arkitektur og udviklerguide.docx](https://github.com/trifork/HAIBA-EPIMIBAimporter/blob/master/doc/Design,%20arkitektur%20og%20udviklerguide.docx)

| Version | Dato | Ændring | Ansvarlig |
| --- | --- | --- | --- |
| 1 | 2014-02-13 | Initielt dokument | Trifork  Aksel Schmidt |
|  |  |  |  |